Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

TÍTULO DEL TFG: Búsqueda y detección de embarcaciones en el mar

TITULACIÓN: Grado de ingeniería en aeronavegación

AUTOR: Azamat Kinzyabaev

DIRECTOR: Sergi Tres Martínez

CO-DIRECTOR: Miguel Valero García

FECHA: 07/07/2019

Título: Búsqueda y detección de embarcaciones en el mar

Autor: Azamat Kinzyabaev

Director: Sergi Tres Martínez

Fecha: 07/07/2019

Resumen

Freeda es un proyecto dirigido por Sergi Tres y tiene como propósito dotar a la oenegé Proactiva Open Arms de drones para las tareas de búsqueda y rescate de personas que quedan a la deriva en el mediterráneo tras el intento fallido de alcanzar las costas europeas.

Para lograr tal propósito, el equipo Freeda de Hemav Foundation, junto con los miembros de Hemav y Proactiva Open Arms, ha trabajado en el desarrollo de un sistema capaz de operar de forma autónoma desde el barco de Proactiva Open Arms. El desarrollo del sistema se ha dividido en diferentes líneas de trabajo que se han asignado a cada miembro del grupo. Sin embargo, casi todas las tareas realizadas por los miembros del equipo han tenido un carácter transversal, ya que se debía integrar todo el trabajo en el diseño final procurando el correcto funcionamiento del sistema en conjunto. Por este motivo, a pesar de que se hace un énfasis en las tareas realizadas por mí, este documento trata de describir el proyecto en su conjunto, puesto que todos los miembros del grupo han tenido que participar de una u otra forma en casi todas las tareas.

En este documento se explicará el desarrollo del proyecto Freeda. Las líneas principales del desarrollo del proyecto han sido: el desarrollo de un sistema de aterrizaje mediante visión artificial, la modificación de la aplicación de control de tierra para acelerar la planificación de misiones de rescate, el desarrollo de los protocolos del dron en el mar para su operación autónoma, y finalmente la implementación de un sistema de procesado de imágenes para la detección de embarcaciones en el mar.

Junto conmigo, en este proyecto han trabajado Francesc Viaplana, del grado en Ingeniería en Vehículos Aeroespaciales de la ESEIAAT, y Eloy Peña, del master’s degree in Aerospace Science and Technology en la EETAC.

Title: Search and detection of vessels in the sea

Author: Azamat Kinzyabaev

Director: Sergi Tres Martínez

Date: 07/07/2019

Overview

Freeda is a project directed by Sergi Tres and its aim is to equip the ONG Proactiva Open Arms with drones for search and rescue tasks of people left adrift in the Mediterranean after the failed attempts to reach the European coasts.

To achieve this purpose, the Freeda team of the Hemav Foundation, together with the members of Hemav and Proactiva Open Arms, have worked on the development of a system capable of operating autonomously from the Proactiva Open Arms ship. The development of the system has been divided into different lines of work, each of which was assigned to a member of the group. Notwithstanding, almost all the tasks carried out by the team members have a transversal nature, hence all the work had to be integrated into the final design, ensuring the correct functioning of the system as a whole. For this reason, although the tasks performed by me have a special emphasis, the present document pursues to describe the entire project, since all the members of the group have participated, in one way or another, in almost all the tasks.

This document will explain the development of the Freeda project. The main lines of development of the project have been: the development of a landing system using artificial vision technology, the modification of the ground control application to speed up the planning of rescue missions, the development of drone protocols at sea for its autonomous operation, and finally the implementation of an image processing system for the detection of vessels at sea.

Francesc Viaplana, from ESEIAAT's Aerospace Vehicle Engineering degree, and Eloy Peña, from EETAC's master's degree in Aerospace Science and Technology and I have worked on this project.

Índice

Introducción

Hasta la fecha de hoy el número de muertos en el Mediterráneo supera ya los 35 000. En Europa se mira con mucho escepticismo hacía la oleada de migrantes que intentan entrar en el continente y las políticas en la mayoría de los países son restrictivas con estos movimientos de personas. Hartos de ver esta situación, la oenegé Proactiva Open Arms lleva a cabo misiones de búsqueda y rescate de gente que se lanzan al mar intentando huir del hambre y las guerras que han arrasado sus hogares. Desde el equipo de Hemav Foundation, Sergi Tres intenta ayudar a Proactiva Open Arms en su labor de búsqueda y rescate de inmigrantes. De esa sinergia nació el proyecto Freeda que tiene como objetivo dotar de drones autónomos a la tripulación del barco Open Arms Bilbao. Estos drones permitirían aumentar el rango y la velocidad de búsqueda de embarcaciones a la deriva que cruzan el Mediterráneo. Además, este sistema está pensado para aumentar la seguridad de los rescatadores, ya que en muchas ocasiones se han envuelto en situaciones violentas con los guardacostas libios, que no dudan en abrir fuego hacia cualquier persona que consideren hostil.

El equipo de desarrollo de Freeda está formado por mí mismo, Francesc Viaplana, del grado en Ingeniería en Vehículos Aeroespaciales de la ESEIAAT, y Eloy Peña, del master's degree in Aerospace Science and Technology en la EETAC. Juntos hemos trabajado en el desarrollo de un dron capaz de despegar y aterrizar de forma autónoma en un barco, detectar embarcaciones a la deriva utilizando una cámara y software de visión artificial y capaz de operar de forma autónoma realizando misiones de búsqueda en el mar. También se ha modificado la aplicación de estación de control de tierra de Hemav, el Hemav Planner, que está basado en el software de código libre de ArduPilot.

En el primer capítulo se contextualiza el proyecto dentro de la operativa de Proactiva Open Arms y se describen las tareas planificadas para el desarrollo del proyecto. También se explica la asignación de las tareas a los miembros del equipo Freeda.

En el segundo capítulo se explica de forma descriptiva en qué consiste cada tarea, cómo se ha desarrollado, cuáles han sido las distintas líneas de desarrollo para completar los objetivos de la tarea, así como formas alternativas de solucionar los problemas.

El capítulo 3 es una descripción de las diferentes plataformas de drones que podrían utilizarse en este proyecto, cuáles son las características de cada uno, así como las ventajas y desventajas, y finalmente cuál ha sido la plataforma elegida y por qué motivo.

En el capítulo 4 se explica más detalladamente la tarea de modificar la estación de control de tierra. Se analiza su diseño para poder entender mejor su funcionamiento y posteriormente poder adaptarla a las necesidades de este proyecto. Finalmente se muestra una prueba realizada con el software modificado para verificar el correcto funcionamiento.

El quinto capítulo son dos estudios de la viabilidad para la implementación de tecnologías que funcionan con señales por satélite. El primero es el estudio de un sistema para el aterrizaje basado en posicionamiento por satélite diferencial (DGNSS). El segundo es una propuesta de modelo de arquitectura basada en radioenlace por satélite.

En el capítulo 6 se explicará cuáles fueron las pruebas que se realizaron para comprobar el funcionamiento del sistema, primero en las instalaciones de Hemav y después en Burriana (Castellón) con el barco Open Arms Bilbao. También se explican cuáles fueron las modificaciones realizadas después de las pruebas.

Por último, el capítulo 7 es un resumen del trabajo con las conclusiones extraídas y una breve explicación de las futuras líneas de trabajo.

CAPÍTULO 1. Proyecto Freeda

1.1. Vehículo aéreo no tripulado: etimología

La denominación *vehículo aéreo no tripulado* o VANT proviene del inglés *unmanned aerial vehicle*, de siglas UAV. Es también muy usada la denominación *sistema aéreo no tripulado*, del inglés *unmanned aerial system* y de siglas UAS.

Más extendido es el término *dron*, recogido en la 23.ª edición del Diccionario de la lengua española, derivado por asimilación del inglés *drone*, que literalmente significa zángano, siendo su forma plural regular en español *drones*. Al tratarse de una adaptación al español, no es preciso destacarla con cursivas ni comillas. Con este término se designan diversos tipos de vehículos aéreos no tripulados. En una primera etapa, este término aludía a aparatos básicamente de uso militar y con aspecto similar al de un avión, por lo que se extendió como alternativa al término procedente del inglés la expresión *avión no tripulado*, que puede considerarse adecuada en muchos casos. No obstante, en los últimos tiempos han surgido otros vehículos que apenas guardan semejanza con los aviones. Para ellos pueden emplearse expresiones más genéricas como *vehículos aéreos no tripulados* o *robots voladores*, según los casos. Si, en todo caso, se prefiere utilizar el término original en inglés *drone* (terminado en -e), lo apropiado es resaltarlo en cursiva o entre comillas por tratarse entonces de un extranjerismo no adaptado.

Otras alternativas usadas por las fuentes son *aeronave no pilotada* o *aeronave no tripulada* y **RPAS**, que proviene de las siglas en inglés *Remotly Piloted Aircraft System* y es aceptado por la Organización de Aviación Civil Internacional.

1.2. Situación actual en el mediterráneo

La guerra de Siria junto con otros conflictos en África y Oriente Medio, además de la situación de pobreza e inestabilidad política de los países de estas regiones, ha provocado la movilización de millones de personas que huyen en busca de una vida mejor. Muchos de ellos tratan de llegar a Europa por mar, actualmente cruzando el Mediterráneo central, causando miles de víctimas cada año.

Proactiva Open Arms trabaja a diario en la zona socorriendo y salvando a aquellas personas que se lanzan al mar con la única esperanza de ser rescatados.

1.3. Qué es el proyecto Freeda

Este proyecto nace de la colaboración entre Hemav Foundation y Proactiva Open Arms. Por ello, ante todo contextualizaré quienes son los responsables de este proyecto.

Hemav Foundation es una entidad privada sin ánimo de lucro que desarrolla proyectos en el ámbito social, con el objetivo de demostrar y promover el compromiso de la tecnología UAV con la sociedad y con el ser humano.

Proactiva Open Arms es una organización no gubernamental y sin ánimo de lucro cuya principal misión es rescatar del mar a los refugiados que llegan a Europa huyendo de conflictos bélicos, persecución y pobreza.

Hemav Foundation quiere colaborar con las tareas de rescate de Proactiva Open Arms introduciendo drones de última tecnología que mejoren y faciliten la búsqueda de barcos a la deriva. El equipo del Proyecto Freeda trabaja en el desarrollo de las plataformas y de las aplicaciones de software necesarias para el funcionamiento autónomo de los UAV, así como el procesado automático de las imágenes captadas. El objetivo es dotar a Proactiva open Arms de 2 drones preparados para distintas tareas según la situación.

El primer dron es el Dron cautivo, preparado para volar conectado a un cable de alimentación, logrando así una autonomía infinita. Realiza vuelos autónomos a poca altitud desde las lanchas de la ONG, con el objetivo de optimizar el tiempo de búsqueda de barcos al ofrecer una vista aérea que permite detectarlos a más de 10 kilómetros.

El segundo dron es el Dron de inspección, capaz de volar largas distancias a gran velocidad de forma totalmente autónoma. Realiza barridos en el lugar indicado captando imágenes que son procesadas en el mismo dron. Cuando detecta un objetivo de interés, las imágenes son mandadas a la tripulación de la ONG para que actúe de inmediato.

Las principales ventajas de este sistema de inspección son:

- **Aumento del área inspeccionada**. Capacidad de controlar un espacio más amplio que el actual, realizando vuelos con constancia en zonas no observadas.
- **Reducción del tiempo de detección**. Los drones permitirán detectar embarcaciones con mayor rapidez y dar avisos a tiempo para permitir la actuación de los socorristas.
- **Reducción del peligro de la tripulación**. Los socorristas de Proactiva Open Arms se ven cada vez más amenazados por los guardacostas libios, quienes han llegado a disparar sus armas en múltiples ocasiones.

1.4. Distribución de tareas

En HEMAV Foundation hay cuatro personas encargadas del desarrollo de este proyecto. El responsable del proyecto es Sergi Tres. Bajo su supervisión trabajamos 3 estudiantes de la UPC, Francesc Viaplana, del grado en Ingeniería en Vehículos Aeroespaciales de la ESEIAAT; Azamat Kinzyabaev, del grado en Ingeniería de Aeronavegación, de la EETAC, y Eloy Peña, del master's degree in Aerospace Science and Technology en la EETAC.

Para distribuir las tareas del proyecto primero hagamos una lista de acciones que deberá realizar el dron para llevar a cabo la misión.

El UAV deberá:

- Despegar desde el barco
- Dirigirse hacia las coordenadas que se le han indicado
- Realizar las fotografías dentro del polígono que se le ha indicado
- Detectar las embarcaciones a la deriva que encuentre en las fotos
- Enviar una alarma con las coordenadas y la fotografía de la embarcación encontrada
- Realizar una búsqueda más precisa en forma de espiral con el objetivo de encontrar otras embarcaciones cercanas
- Continuar la misión
- Volver a la posición del barco
- Aterrizar en el barco

De acuerdo con los puntos establecidos hemos dividido el proyecto en 4 Work Packages.

1. WP1: Crear la misión inicial. Modificar la plataforma de tierra (Mission Planner) para facilitar la labor a los operarios del Open Arms Bilbao. La misión deberá crearse fácil y deberá incluir la ruta desde el barco a las coordenadas indicadas, realizar una misión de búsqueda en torno a ese punto y volver a la posición del barco una vez finalice.
2. WP2: Protocolos del UAV durante la misión. En este punto se deberá programar al dron para que, en el caso que detectara una embarcación a la deriva, calcule las coordenadas del barco a la deriva, su rumbo y su velocidad, envíe las coordenadas y las fotos al Open Arms Bilbao, comience la misión en espiral para buscar barcos cercanos al ya encontrado, y finalmente prosiga con la misión principal.
3. WP3: Detección de embarcaciones con Visión Artificial. Montar una Red Neuronal capaz de reconocer embarcaciones en las imágenes tomadas por el dron.
4. WP4: Aterrizaje del UAV. Diseño de un sistema de aterrizaje capaz de implementarse en el barco de forma segura.

Según la complejidad de las tareas a realizar y las aptitudes de cada uno hemos decidido dividir el proyecto de la siguiente forma:

- Eloy Peña: WP3 y WP4
- Azamat Kinzyabaev: WP1 y WP4
- Francesc Viaplana: WP2

CAPÍTULO 2. Tareas del proyecto

2.1. Preparar la estación de tierra (WP1)

La estación de control de tierra sirve para planificar la misión, monitorizar la actividad del UAV durante la misión, interactuar con el UAV, en definitiva, para controlar el dron remotamente. El objetivo de esta tarea es entregar a Open Arms una aplicación que les sirva como estación de tierra para las misiones de salvamento marítimo. Esta aplicación se basará en el proyecto de código libre Mission Planner de ArduPilot. En Hemav ya existe una versión adaptada de Mission Planner denominada Hemav Planner que reduce algunas complejidades de la aplicación y las adapta a las necesidades de sus proyectos y usuarios.

La aplicación de estación de tierra para salvamento marítimo será desarrollada a partir de Hemav Planner. Por ello primero analizaremos la aplicación para entender mejor sus funcionalidades y limitaciones.

2.1.1. Limitaciones de la estación de tierra existente de cara a este proyecto

La principal limitación es la complejidad de la aplicación. No es una aplicación intuitiva y es complicado navegar por ella si no estás familiarizado. Hay muchos aspectos presentes en el programa que son totalmente prescindibles para esta aplicación. Además, se precisan conocimientos mínimos de manejo de drones o haber trabajado previamente con drones para entender bien qué acciones se deben realizar para poder sacarle provecho a esta herramienta.

Otra limitación es el tiempo que se tarda en preparar una misión. El proceso es largo y bastante manual, ya que debe satisfacer numerosos tipos de misiones y es por ello por lo que está poco automatizado.

2.1.2. Cambios para la nueva estación de tierra

Para poder adaptar la aplicación se debe tener en cuenta que

1. La misión debe crearse rápidamente ya que el tiempo es crucial en estas misiones.
2. Dotar de cierta versatilidad al sistema. El polígono debe poder modificarse una vez creado. El usuario debe poder rotar el polígono, darle forma según convenga a la misión.
3. La dirección del viento influye en la autonomía, que es un factor clave en este proyecto.
4. Intentar cubrir el máximo área posible teniendo en cuenta la autonomía del UAV.
5. Usar el mínimo número de acciones para crear y ejecutar la misión.

6. Darle al usuario la información necesaria para la misión de rescate, evitando información redundante y poco práctica para disminuir la complejidad de la interfaz de usuario.

Modificaciones:

- La primera ventana de selección de plataforma se suprime ya que la plataforma con la que se va a operar siempre va a ser la misma. Por lo tanto, habrá que modificar el código de forma que venga ya establecido el tipo de UAV con el que se va a operar.
- El método para crear los polígonos de las misiones cambiará. El polígono se generará entorno a unas coordenadas de latitud y longitud que el usuario entrará. El área del polígono dependerá de la autonomía del dron y de la distancia que tenga que recorrer.
- Habrá que realizar los vuelos durante la misión en perpendicular al viento ya que de esa forma se optimiza la autonomía. Por lo tanto, habrá que habilitar una forma de introducción de la dirección del viento.
- Hay que automatizar las tareas preparativas para el despegue del dron reduciendo el número de comandos que el usuario deberá introducir para comenzar la misión.
- Habrá que eliminar de la pantalla toda aquella información que no sea indispensable para la realización de la misión.

2.2. Diseño de un sistema de detección de embarcaciones (WP2)

La forma que se ha elegido para detectar las embarcaciones en alta mar es por ‘Computer Visión’ o Visión Artificial. Para poder implementarlo la principal librería que requiere el sistema es OpenCV. OpenCV provee una infraestructura para aplicaciones de visión artificial y tiene más de 2500 algoritmos, que incluye algoritmos de machine learning y de visión artificial para usar. Estos algoritmos permiten identificar objetos, caras, clasificar acciones humanas en vídeo, hacer tracking de movimientos de objetos, extraer modelos 3D, encontrar imágenes similares, eliminar ojos rojos, seguir el movimiento de los ojos, reconocer escenarios, etc.

Estos algoritmos se ejecutarán en una Raspberry Pi3 que llevará una cámara integrada de 8 megapíxeles.

Se requiere realizar una serie de procesados de imagen para poder implementar esta librería. El procesado consta de los siguientes pasos:

1. Conversión a escala de gris: La mayoría de operación de procesado trabajan con imágenes en escala de grises
2. Suavizado mediante filtro Gaussian Blur: Realiza convoluciones con un kernel especificado que deberá ser ajustado dependiendo de la altura de vuelo
3. Thresholding adaptativo utilizando la media de los pixeles cercanos: Gracias a que el valor del thresholding depende de los pixeles cercanos se mitigan los reflejos y la espuma de las olas

4. Transformaciones morfológicas de erosión y dilatación: Se efectúa una erosión para eliminar ruido y una dilatación para recuperar la forma original
5. Detección de contornos y clasificación por jerarquías: Se detectan los bordes que encierran algún área y las relaciones de parentesco entre ellos
6. Eliminación de los contornos interiores: Haciendo uso de las jerarquías se eliminan los contornos hijos reduciendo así el tiempo de cálculo
7. Filtrado de áreas por tamaño: Utilizando el valor esperado de pixeles se eliminar las áreas fuera del rango estimado
8. Envolvente convexa: Proporciona una mejor delimitación del área detectada reparando pequeños fallos en la detección del contorno
9. Cálculo de centroides: Determinado los momentos de las áreas se calcula el centro de masas que es devuelto como coordenadas de los objetivos

La principal labor en esta tarea es adaptar los algoritmos de la librería Open CV y desarrollar un código que realice el procesado de imágenes comentado anteriormente. El código está escrito en Python.

Para entender mejor estas técnicas de procesado de imagen veamos un caso práctico en el que se detecta un navío

Lo primero que necesitamos hacer es parametrizar y ubicar al dron en relación con las distancias que aparecen en las imágenes captadas. Para ello se debe calcular el área que se va a fotografiar, ya que el tamaño de los objetos detectados es importante.

Tabla 2.1. Variables para el cálculo de la huella fotográfica.

Variables	
Xsensor	Ancho del sensor [mm]
Ysensor	Altura del sensor [mm]
Lf	Longitud focal de la lente [mm]
h	Altitud del dron [m]
Xgimbal	Eje x del ángulo de gimbal
Ygimbal	Eje y del ángulo de gimbal

Ancho del campo de visión de la cámara [º]:

$$\alpha_{ancho} = 2\,tan^{-1}\frac{X_{sensor}}{2LF}$$

Altura del campo de visión de la cámara [º]:

$$\alpha_{altura} = 2\,tan^{-1}\frac{Y_{sensor}}{2LF}$$

Distancia desde la cámara hasta la parte superior del campo de visión [m]:

$$D_{sup} = h\,tan(Y_{gimbal} - \frac{1}{2}\alpha_{altura})$$

Distancia desde la cámara hasta la parte inferior del campo de visión [m]:

$$D_{inf} = h\ tan(Y_{gimbal} + \frac{1}{2}\alpha_{altura})$$

Distancia desde la cámara hasta la parte izquierda del campo de visión [m]:

$$D_{izq} = h\,tan(X_{gimbal} - \frac{1}{2}\alpha_{ancho})$$

Distancia desde la cámara hasta la parte derecha del campo de visión [m]:

$$D_{der} = h\tan\left(X_{gimbal} + \frac{1}{2}\alpha_{ancho}\right)$$

Altura de la huella fotográfica [m]: $H = D_{der} - D_{izq}$

Ancho de la huella fotográfica [m]: $W = D_{sup} - D_{inf}$

Una vez se conocen las dimensiones de la imagen se puede establecer una relación entre los pixeles de la imagen y distancias reales.

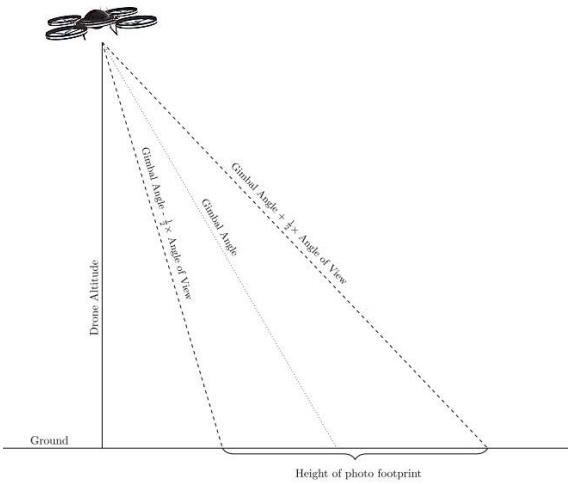

Figura 2.1. Esquema de la visión de la cámara del dron y los parámetros para el cálculo de la huella fotográfica.

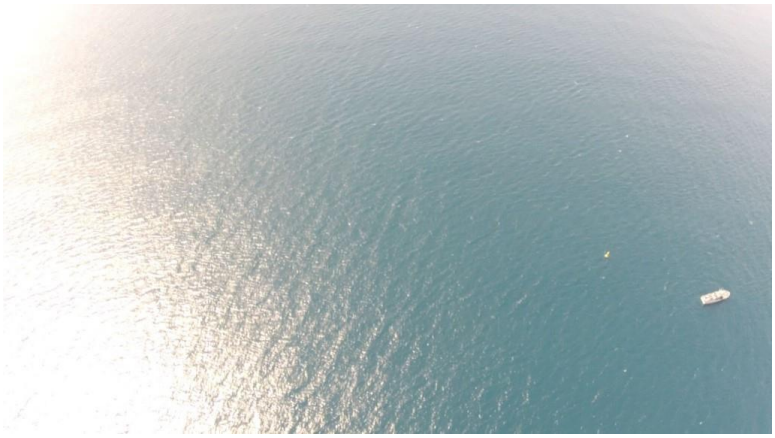

Figura 2.2. Fotografía de una embarcación en el mar desde una perspectiva aérea.

Para poder detectar la embarcación en esta imagen hay que aplicar los procesados de imagen comentados anteriormente. Lo primero que se hace es pasar esta imagen a escala de grises para posteriormente aplicarle la técnica de 'thresholding'. El 'thresholding' consiste en coger el tono de gris medio de la

imagen y, convertir a negro los pixeles con tonos que estén por encima de la media y convertir a blanco los que estén por debajo.

Después se realizan las operaciones morfológicas que consiste en dos técnicas. Primero realizas una erosión-dilatación. Esto permite quitar el ruido de la imagen, primero reduciendo el tamaño de los objetos encontrados en la imagen y después engrosándolos.

Después podemos buscar contornos en la imagen y establecer formas definidas. La foto procesada queda de esta forma:

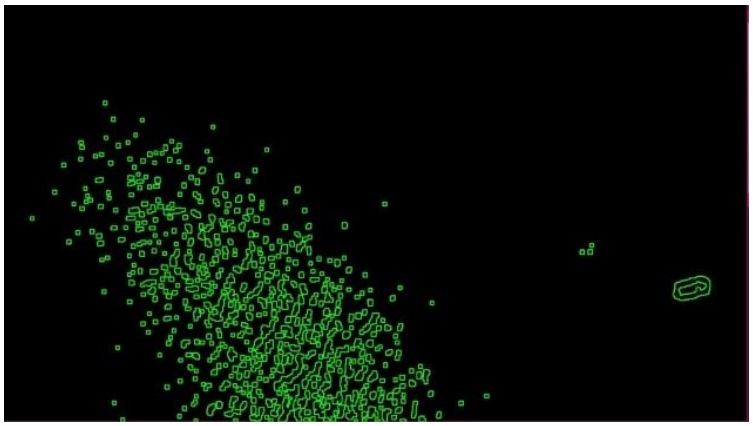

Figura 2.2. Imagen después del procesado de operaciones morfológicas.

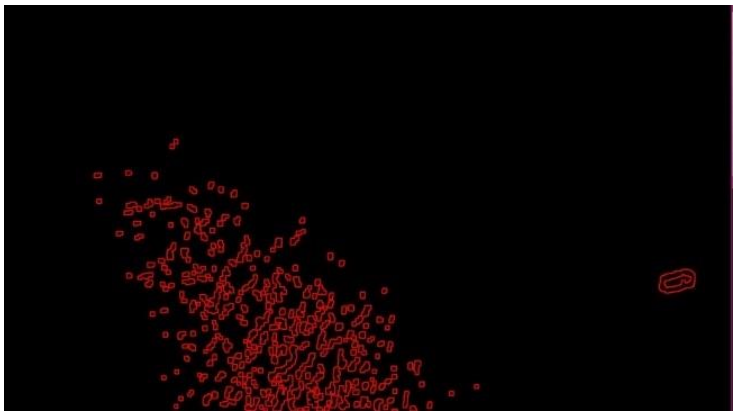

Figura 2.3. Imagen después del filtrado por tamaños.

El paso siguiente es indicar qué tamaños son los que nos interesan. Esto lo podemos hacer gracias a la 'Footprint' de la cámara calculada previamente. El algoritmo calcula las áreas de los contornos de la imagen y guarda los que estén dentro de los límites indicados.

Posteriormente se ejecuta un algoritmo de 'Clustering' que agrupa las formas dependiendo si forman un conjunto unido o elementos independientes. Básicamente este filtro crea una función para la imagen y la clasifica entre señal y ruido.

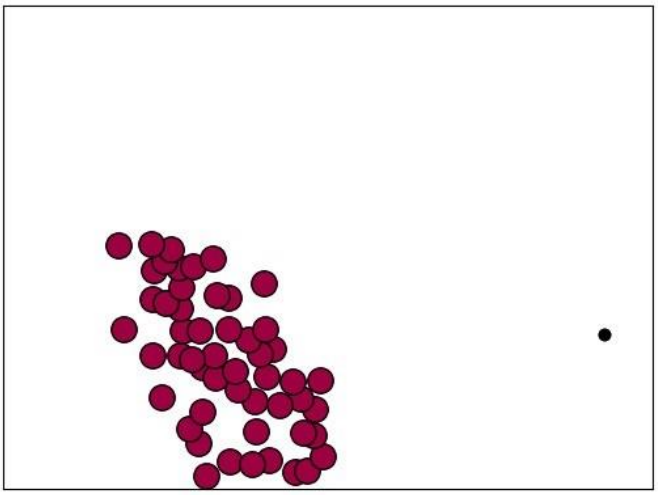

Figura 2.4. Agrupación de elementos en la imagen según si forman un grupo u objeto individual.

El barco es considerado ruido ya que es un objeto individual y no tiene ninguna relación con otros objetos que puedan aparecer en la imagen. Los contornos correspondientes al ruido de la función son elegidos como objetos de interés.

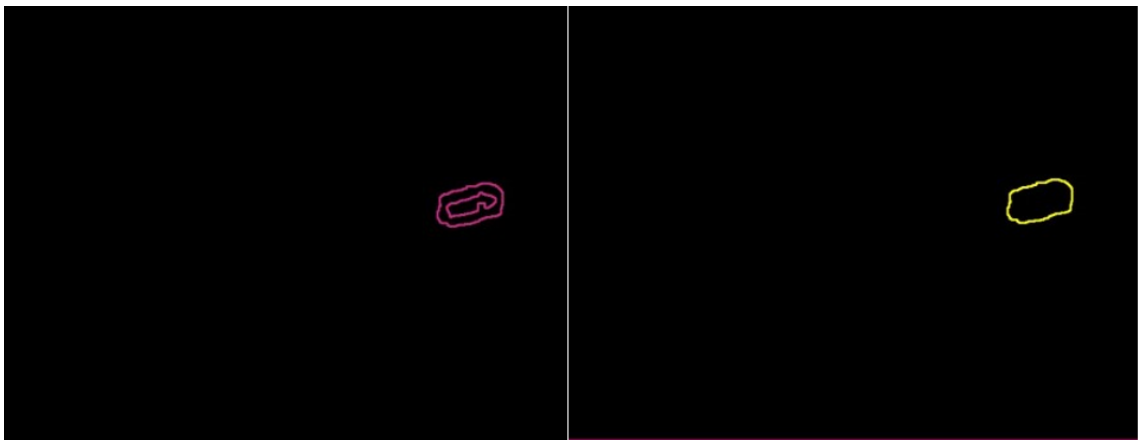

Figura 2.5. Imagen antes y después del procesado por jerarquías.

El último paso es aplicar una jerarquía de contornos. Los contornos que estén dentro de otros serán considerados un mismo objeto.

De esta forma nos queda solo un contorno en la imagen procesada que podemos relacionar con el barco que queremos detectar.

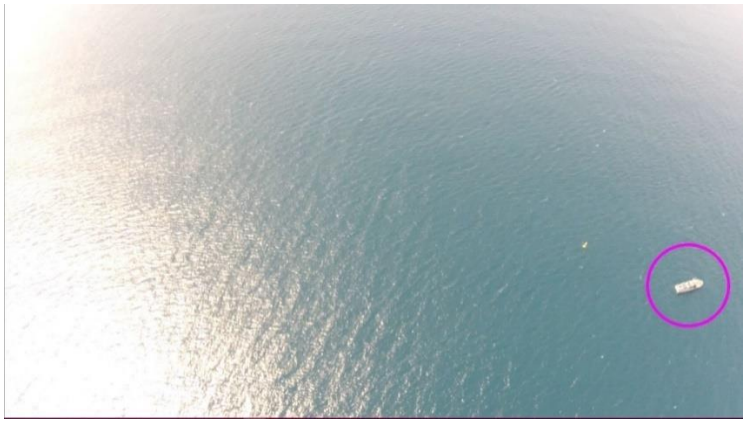

Figura 2.6. Imagen con la embarcación detectada.

Una vez comprendida la técnica de procesado de imagen para la detección, hay que asegurar que el software es fiable y funciona para todos los casos de luminosidad, color y tipo de embarcaciones. El correcto funcionamiento del sistema se analiza en el capítulo de las pruebas del sistema.

2.3. Diseño de los protocolos del dron en el mar (WP3)

En esta sección se explican los protocolos que se deben implementar para que el UAV opere de forma autónoma en la misión de búsqueda de navíos en el mar.

El procedimiento comienza cuando se crea la misión principal desde la estación de control de tierra. A continuación, el UAV despega desde el barco y se dirige hacia el polígono donde está definida la misión de búsqueda.

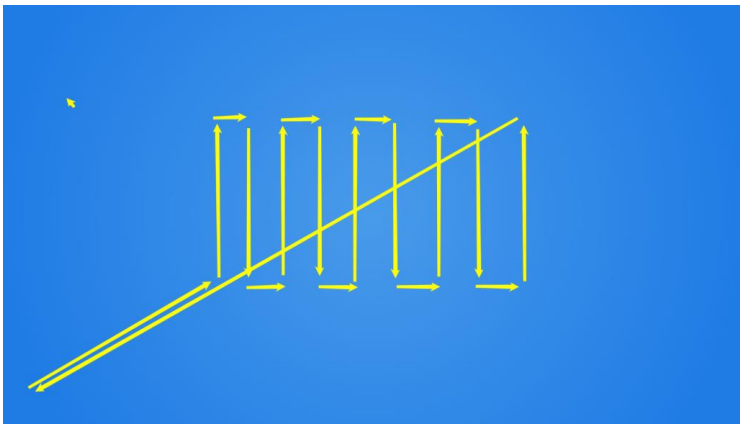

Figura 2.7. Representación esquemática de la ruta que seguirá el dron durante la misión de búsqueda.

Una vez llegue al punto de entrada en el polígono el dron comienza a hacer fotografías de la superficie del mar y el programa de detección de embarcaciones empieza a correr. A continuación,

- Si se detecta una embarcación en la imagen capturada el dron realiza varias acciones
  - Calcula las coordenadas de la embarcación comparando sus coordenadas con la posición de los pixeles de la imagen, la huella fotográfica calculada y la inclinación de la cámara.
  - Crea un área de estimación alrededor de las coordenadas de la embarcación que ha detectado.
  - Comienza a orbitar esas coordenadas formando una espiral realizando más fotografías.
- Si durante la órbita en espiral el algoritmo de detección vuelve a detectar una embarcación
  - Calcula sus coordenadas y las comprueba si están dentro del área de estimación calculado previamente
    - Si está dentro del área de estimación es la misma embarcación que se detectó antes
    - Si está fuera del área de estimación es otra embarcación diferente
    - Si es una embarcación diferente, se repite el proceso de la espiral y del área de estimación para esta nueva embarcación
- Con las coordenadas de las embarcaciones y los momentos en que se tomaron, el ordenador a bordo del dron calcula el rumbo y la velocidad de las embarcaciones a la deriva y las envía al barco de Open Arms.
- Si después de detectar una embarcación el dron lleva un tiempo mayor que un tiempo T sin detectar vuelve a la misión principal.

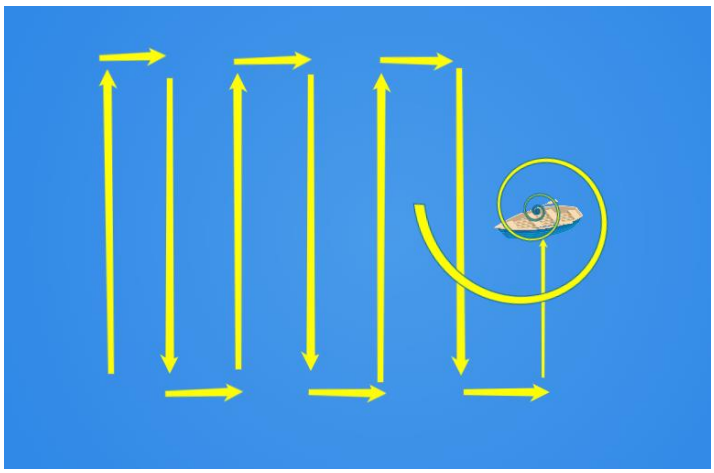

Figura 2.8. Representación esquemática de la trayectoria en espiral del UAV cuando detecte una embarcación.

2.4. Diseño de un sistema de aterrizaje (WP4)

Uno de los retos más importantes de este proyecto es conseguir el dron aterrice en el barco Open Arms Bilbao de forma segura. Para entender la problemática de esta tarea primero explicaré cuales son los métodos que se usan actualmente para aterrizar un dron.

Normalmente se usan los sensores inerciales y GPS para enviar al dron a unas coordenadas donde se procede al protocolo de aterrizaje. El descenso final se hace mediante control remoto donde el piloto aterriza el dron manualmente. Para los aterrizajes autónomos se utilizan métodos como GPS diferencial, que mejoran notablemente el nivel de precisión de los sensores. También se pueden integrar con otras radioayudas como antenas directivas o sistemas de antenas que emiten señales de distintas fases que son capaces de dirigir un vector hacia una localización. Otro de los métodos muy utilizados en la industria de los UAV es aterrizaje mediante visión artificial. Con una cámara los drones son capaces de localizar y dirigirse hacia objetivos donde puedan realizar un aterrizaje seguro.

2.4.1. Factores de riesgo

Para poder implementar un sistema para aterrizar el UAV por coordenadas hace falta un buen sistema de posicionamiento por GPS. Estos equipos normalmente son caros y además hay que considerar la dificultad añadida de que el barco nunca se encuentra sobre unas mismas coordenadas, sino que va a la deriva y se mueve con la marea. La opción de controlar al dron con las imágenes procesadas por visión artificial tampoco es sencilla ya que dirigirlo a un objetivo que se está balanceando es complicado.

También debemos tener presente los siguientes factores antes de empezar a planificar la forma de aterrizaje:

- El barco está en misión de rescate y es muy probable que haya gente en la cubierta del barco cuando el UAV tenga que aterrizar.
- Las dimensiones del barco no son grandes.
- El barco está continuamente en movimiento, sobre todo cuando las condiciones meteorológicas son adversas.

La fase de aterrizaje no puede salir mal porqué eso supondría la posible pérdida del UAV o alguna de sus componentes, algún daño en el barco y, por encima de todo, alguien podría resultar herido.

2.4.2. Aterrizaje mediante visión artificial

Esta es una de las opciones que más gusta ya que sólo hay que equipar al dron con una cámara adicional por lo que esta opción resulta la más económica.

Esta solución consiste en la colocación de una red con un marco verde en el barco. El sistema de detección reconoce el marco verde y localiza su centro. Mediante la localización de los pixeles de la imagen correspondientes al marco y a su respectivo centro y la inclinación de la cámara podemos traducir esta información en una respuesta de los actuadores encargados del control de la aeronave y dirigirla contra la red. De esta forma el dron, que llevará incorporado un pequeño gancho en el morro, quedará atrapado en la red.

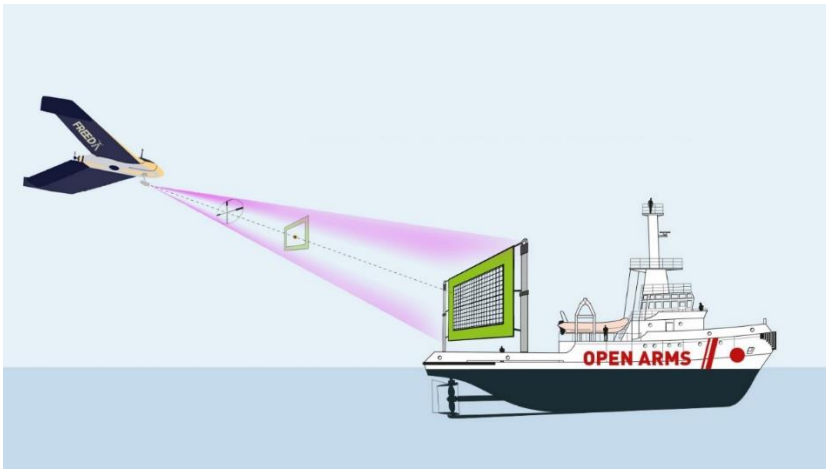

Figura 2.9. Representación esquemática de la red instalada en la popa del barco y la aproximación del UAV en la fase de aterrizaje.

Para poder implementar este sistema el dron se dirigirá a unas coordenadas donde creerá que estará la red. Creemos que el sistema de posicionamiento convencional por GPS será suficiente como para aproximarse al barco con suficiente precisión como para poder detectar el marco de la red y poder ejecutar el protocolo. Se realizarán las correspondientes pruebas para determinar la distancia de detección.

2.4.3. **Aterrizaje mediante GNNS Diferencial**

Un sistema de posicionamiento global diferencial (DGPS) es una mejora del Sistema de posicionamiento global que proporciona una precisión de ubicación mejorada, en el rango de operaciones de cada sistema, desde la precisión GPS nominal de 15 metros hasta aproximadamente 1-3 cm en el caso de las mejores implementaciones.

Cada DGNSS utiliza una red de estaciones de referencia terrestres fijas para transmitir la diferencia entre las posiciones indicadas por el sistema de satélites GPS y las posiciones fijas conocidas. Estas estaciones emiten la diferencia entre las pseudodistancias del satélite medido y las pseudodistancias reales calculadas internamente, y las estaciones receptoras pueden corregir sus

pseudodistancias en la misma cantidad. La señal de corrección digital generalmente se transmite localmente a través de transmisores terrestres de menor alcance.

Para poder implementar este sistema en nuestro proyecto necesitaríamos una señal de referencia fija capaz de tener un alcance suficiente para poder operar en el mar. La opción más viable seria instalarlo en el mismo barco, pero en ese caso ya no estaríamos hablando de una referencia fija sino de una referencia móvil.

La viabilidad de implementar un sistema de posicionamiento global diferencial con una referencia móvil se someterá a un estudio en capítulos posteriores.

CAPÍTULO 3. El dron

3.1. Determinación de la plataforma para la misión

Generalmente podemos definir dos tipos de vehículos aéreos motorizados, los de ala fija y ala giratoria. Dentro de los vehículos de ala giratoria podemos distinguir los helicópteros y los multirrotores. Las ventajas e inconvenientes de cada plataforma las determina el tipo de operaciones que se pretenda realizar, dependiendo de si nos interesa aprovechar el alcance, la autonomía, transportar gran carga de pago, manejabilidad, etc.

En esta sección se revisa cuáles son las diferentes plataformas de drones con las que se puede operar, pero también qué hardware requieren y cómo se establece el sistema de control e información de la plataforma.

3.1.1. Multirrotor

Los multirrotores son la herramienta más extendida actualmente y la que todos al pensar en drones tenemos en nuestra mente. Proporciona una gran versatilidad y eficacia en las operaciones por su simpleza a la hora de ser pilotados y por la velocidad de montaje. Es una plataforma estable por naturaleza, debido a que los motores se encuentran a la misma distancia del centro de gravedad de la aeronave. A la hora de elegir un multirrotor u otro, debemos de tener en cuenta que cuantos más brazos tengamos más estabilidad y más seguridad, por contrapartida tendremos más propulsión y consumo. Es importante destacar que un multirrotor de media no suele superar los 15 minutos de vuelo, lo que representa un gran impedimento para muchas operaciones.

Figura 3.1. UAV multirrotor HIR9.

3.1.2. Helicóptero

Los helicópteros son la herramienta más versátil a la hora de realizar todo tipo de operaciones. Poseen la ventaja de realizar vuelo estacionario además de gran carga de pago y mucha autonomía. Esto es gracias a que sólo posee un motor

y una hélice de gran tamaño. El consumo energético en comparación con un cuadricóptero se reduce alrededor de un 25%. El helicóptero es mucho más eficiente aerodinámicamente que un multirrotor, ya que el helicóptero funciona a revoluciones fijas de motor gracias al paso variable de las hélices, mientras que el multirrotor varía las revoluciones del motor para mantenerse estable. Sin embargo, los helicópteros son bastante complejos a nivel mecánico, lo que nos obliga a tener que estar constantemente ajustándolo para que nos ofrezca un vuelo óptimo. También es bastante complicado a la hora de ser pilotado, y dominarlos suele llevar bastantes años de práctica.

Figura 10. Helicóptero no tripulado Airbus VSR 700.

3.1.3. **Ala fija**

El ala fija es el claro ganador en lo que a autonomía se refiere. Es la plataforma perfecta para trabajos que abarquen una gran extensión de terreno. Es el más eficiente aerodinámicamente hablando ya que, con la configuración adecuada, puede permanecer bastante tiempo sin necesidad de utilizar el motor gracias al planeo. Su principal desventaja son el aterrizaje y el despegue. Es necesario habilitar una gran zona libre de obstáculos para alcanzar la velocidad de despegue o utilizar herramientas como catapultas y tirachinas para acelerarlos. Otras desventajas del ala fija son que posee una capacidad de carga de pago más reducida respecto a un helicóptero y tampoco puede realizar vuelo estacionario. Un dron de ala fija tampoco permite hacer un vuelo estacionario.

Figura 11. Dron de ala fija HP2 desarrollado en HEMAV.

3.1.4. **VTOL**

Existen plataformas mixtas que intentan aprovechar las ventajas de cada una de las anteriores. Un ejemplo son los VTOL. VTOL, de sus siglas en inglés 'Vertical Take Off and Landing', es la capacidad de ciertos vehículos aéreos de realizar vuelo estacionario, despegar y aterrizar verticalmente. Esta clasificación puede incluir una variedad de tipos de aeronaves que incluyen aeronaves de ala fija, así como helicópteros y otras aeronaves con rotores motorizados.

El que nos concierne a nosotros nos referimos al vehículo volador de ala fija VTOL. Estos vehículos permiten operar en lugares con poco terreno para aterrizar o despegar, conservando su gran autonomía. Debido a la propia naturaleza de estas plataformas, el vuelo tipo de operación sería aquel en el que las fases de ascenso y de descenso fueran lo más cortas posibles para poder maximizar la autonomía de vuelo en modo ala fija. Esto es debido a que el consumo en modo multirrotor se considerablemente superior al consumo en modo ala fija.

Un VTOL permitiría despegar desde la cubierta del Open Arms Bilbao sin demasiadas dificultades y puede abarcar gran distancia para poder sacar máximo provecho al dron como herramienta de búsqueda.

Figura 12. Dron VTOL Foxtech Nimbus 1800.

3.1.5. **Elección de la plataforma**

Tras revisar los distintos tipos de plataformas que existen debemos descartar la opción del multirrotor debido a que su autonomía es insuficiente para poder realizar las misiones de rescate en el rango que se requiere. La opción del helicóptero también se descarta debido a su alto coste, además de que la autonomía sigue siendo insuficiente.

Esto nos deja con el VTOL y el ala fija. La ventaja del VTOL es que el despegue desde el barco es muy sencillo. Por contrapartida tiene una autonomía mucho

más limitada que el ala fija. Sin embargo, el parámetro crucial es el alcance de cada plataforma, lo que nos deja con el ala fija. A pesar de que hay que diseñar un sistema de despegue desde el barco que aún no se ha estudiado, el ala fija es el único que se acerca al rango de búsqueda de Proactiva Open Arms. Además, el ala fija es un aparato de menor complejidad que el VTOL, lo cual es otro punto a favor para este tipo de drones, ya que cualquier cosa que se pueda hacer para facilitar las labores de rescate de los operarios se tiene en cuenta. Por esto y por su gran rango operativo el ala fija es la plataforma elegida para este tipo de misiones.

3.2. Hardware a bordo

3.2.1. Raspberry Pi 3

La Raspberry Pi 3 es un ordenador de placa simple de bajo coste que lleva incorporado un Quad-Core de 1.20GHz con una RAM de 1 GB e incluye Wi-Fi y Bluetooth 4.1 sin necesidad de adaptadores. Además, tiene un puerto Ethernet, cuatro puertos USB, un puerto HDMI, una ranura para una memoria SD. Para grabar las imágenes lleva incorporada una cámara RP V2 de 8 Megapíxeles. El software desarrollado para los protocolos del dron, la detección de la red de aterrizaje y el procesado de imágenes para la detección de las embarcaciones se ejecuta en la Raspberry Pi 3.

Tabla 3.1. Características principales del procesador Raspberry Pi 3 B.

Procesador	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz
Memoria	1GB LPDDR2 SDRAM
Conexiones	• 2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE • Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps) • 4 × USB 2.0 ports
Acceso	Cabecera de 40 pines GPIO
Vídeo y sonido	• 1 × full size HDMI • MIPI DSI display port • MIPI CSI camera port • 4 pole stereo output y composite video port
Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode(1080p30); OpenGL ES 1.1, 2.0 graphics

Soporte de tarjeta SD	Formato microSD para cargar el sistema operativo y el almacenamiento de datos.
Voltaje de entrada	• 5 V/2.5 A DC via micro USB connector • 5 V DC via GPIO • Power over Ethernet (PoE)
Temperatura de funcionamiento	0 – 50 °C

Figura 13. Raspberry Pi 3 B con la cámara Raspberry Pi Camera Module V2.

Tabla 2.2. Características principales de la cámara Raspberry Pi Camera Module V2.

Raspberry Pi Camera Module v2	
Peso	3g
Resolución	8 megapíxeles
Sensor	Sony IMX219
Resolución sensor	3280 x 2464 pixeles
Distancia focal	3,04 mm
Área del sensor de imagen	3.68 x 2,76 mm
Campo de visión	62,2 x 48,8 grados

3.2.2. **Controladora**

Para controlar la aeronave se dispone de un módulo Pixhawk2 cube, que incluye el hardware y el software necesario para recoger la información de los sensores y procesar una respuesta para proporcionar las instrucciones que accionan los motores y los actuadores del dron. Podemos considerar este módulo como el piloto automático del UAV ya que se encarga de procesar la información y dirigir el vehículo a las coordenadas programadas. La Pixhawk2 incluye aislamiento de

vibración en dos de las IMU, con una tercera IMU fija como referencia. Los sensores inerciales y el GPS van separadas del resto del ensamblado.

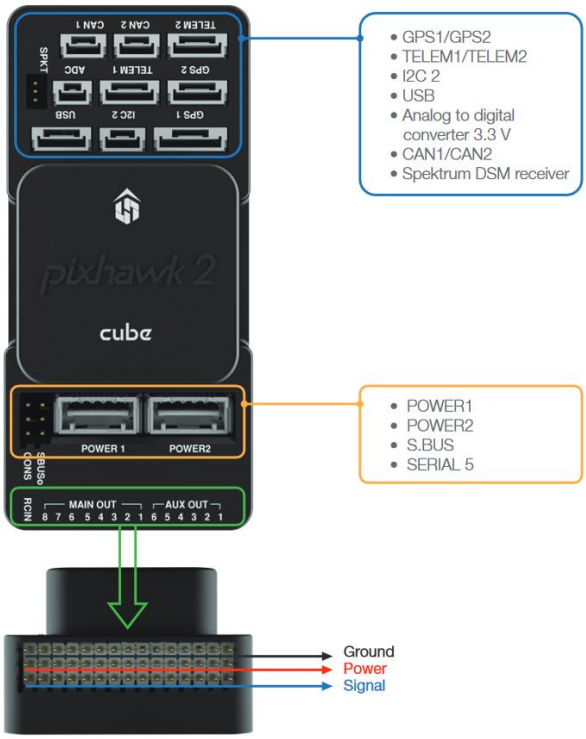

Figura 14. Ilustración del módulo Pixhawk 2 Cube con la descripción de las entradas.

- 32bit STM32F427 Core Cortex-M4F® con FPU
- 168 MHz / 252 MIPS
- 256 KB de RAM
- 2 MB de Flash
- 32 bits STM32F103 co-procesador a prueba de fallos
- 14 salidas PWM / Servo (8 con protección contra fallas y anulación manual, 6 auxiliares, compatible con alta potencia)
- Abundantes opciones de conectividad para periféricos adicionales (UART, I2C, CAN)
- Sistema de respaldo integrado para la recuperación en vuelo y la anulación manual con procesador dedicado y fuente de alimentación independiente (uso de ala fija)
- El sistema de respaldo integra la mezcla, proporcionando un piloto automático consistente y modos de mezcla de anulación manual (uso de ala fija)
- Entradas de alimentación redundantes y failover automático.
- Interruptor de seguridad externo
- Indicador visual principal LED multicolor
- Indicador de audio piezoeléctrico multitono de alta potencia
- Tarjeta microSD para registro de alta velocidad durante largos períodos de tiempo

Para conectar el módulo de telemetría a la controladora se utiliza el puerto TELEM1. Este enlace de datos sirve para transferir todos los datos del vuelo y navegación.

La alimentación se efectúa a través del puerto MICRO-USB.

3.2.3. **Telemetría**

Es la tecnología que permite medir magnitudes físicas y enviarlas de forma remota. Esto permite al operador conocer todas las mediciones de los sensores de a bordo y poder actuar sobre los controles del dron. También es la tecnología que se encarga del envío de datos de detección del UAV al barco de Open Arms. Para desplegar este sistema se debe conectar la estación de tierra al dron a través de un radioenlace. Esto se puede conseguir mediante una antena con un radioenlace directo con el dron o mediante una conexión al satélite.

En nuestro proyecto se utiliza el kit de desarrollo pMDDL2450. El pMDDL2450 es un enlace de datos digitales inalámbrico 2x2 MIMO de alta potencia diseñado para proporcionar conexiones inalámbricas de alto rendimiento en un módulo OEM compacto y robusto para la integración del sistema.

Figura 15. Ilustración del kit de desarrollo pMDDL2450 con todas sus componentes.

El kit de desarrollo incluye:

- (2) pMDDL2450 OEM Modules
- (2) Pico Ethernet Motherboards
- (4) UFL Antenna Cables
- (4) Rubber Ducky Antennas
- (2) Cat5 Ethernet Cables

- (2) RS232 Serial Cables
- (2) 12VDC Power Adapters

Características:

- 2X2 MIMO robusto de 2.4 GHz de operación
- Combinación de Relación Máxima (MRC), Decodificación de Probabilidad Máxima (ML)
- Verificación de Paridad de Baja Densidad (LDPC)
- Hasta 25 Mbps Iperf Rendimiento @ 8 MHz canal (-78 dBm)
- Hasta 2 Mbps Rendimiento de Iperf a un canal de 4 MHz (-102 dBm)
- Extremadamente pequeña huella y muy ligero
- Puerto de comunicación de serie: puertos duales Ethernet 10/100 (LAN / WAN)
- Admite redes punto a punto y punto a multipunto
- Modos de operación Master, Remote y Relay
- Potencia de transmisión total ajustable (hasta 1 W)
- Interfaz a través de la consola local, telnet y navegador web.
- Actualización de firmware inalámbrico remoto y local y local a través de FTP.
- Voltajes de entrada: Digital Voltage = 3.3 V / RF Voltage = 5 V
- Temperatura de funcionamiento: -40 ºC – 85 ºC
- Humedad de funcionamiento: 5 – 95%

Tabla 3.3. Especificaciones del rendimiento óptimo del kit de desarrollo pMDDL2450 según la configuración 2X2 MIMO y según la madulación utilizada.

Especificaciones de rendimiento óptimo			
Modulación	**IPerf Troughput [Mbps]**	**Sensibilidad Óptima MRC [dBm]**	**Potencia Total Transmitida [dBm]**
8 MHz (2X2 MIMO ON)			
BPSK_1/2	3	-99.5	30dBm
QPSK_1/2	5.9	-98	30dBm
QPSK_3/4	8.8	-96	30dBm
16QAM_1/2	11.6	-92	30dBm
16QAM_3/4	17.1	-90	30dBm
64QAM_2/3	22.8	-85	30dBm
64QAM_3/4	25.5	-83.5	30dBm
64QAM_5/6	27.8	-81	30dBm
4 MHz (2X2 MIMO ON)			
BPSK_1/2	1.51	-102.5	30dBm
QPSK_1/2	2.98	-101	30dBm
QPSK_3/4	4.4	-99	30dBm
16QAM_1/2	5.8	-95.5	30dBm
16QAM_3/4	8.6	-93	30dBm
64QAM_2/3	11.4	-88	30dBm

64QAM_3/4	12.8	-86	30dBm
64QAM_5/6	14	-83.5	30dBm
8 MHz (2X2 MIMO OFF)			
BPSK_1/2	3	-96.5	30dBm
QPSK_1/2	5.8	-95	30dBm
QPSK_3/4	8.6	-93	30dBm
16QAM_1/2	11.5	-89	30dBm
16QAM_3/4	16.9	-87	30dBm
64QAM_2/3	22.2	-82	28dBm
64QAM_3/4	24.7	-80.5	28dBm
64QAM_5/6	27.4	-78	27dBm
4 MHz (2X2 MIMO OFF)			
BPSK_1/2	1.5	-99.5	30dBm
QPSK_1/2	2.9	-98	30dBm
QPSK_3/4	4.3	-96	30dBm
16QAM_1/2	5.7	-92.5	30dBm
16QAM_3/4	8.4	-90	30dBm
64QAM_2/3	11.3	-85	28dBm
64QAM_3/4	12.5	-83	28dBm
64QAM_5/6	14	-80.5	27dBm

CAPÍTULO 4. Desarrollo del Hemav Planner

4.1. Organización del Hemav Planner

Hemav Planner es la estación de control de tierra de Hemav construida mediante plugins que se relacionan con Mission Planner para agregarle funcionalidades nuevas, pero sobre todo para reducir el tamaño de la aplicación y adaptarla a las necesidades de los clientes de Hemav. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de una API. La mayor parte del código de esta aplicación está escrita en C# (92%), aunque también tiene partes en Python (3,7%) y JavaScript (3,4%).

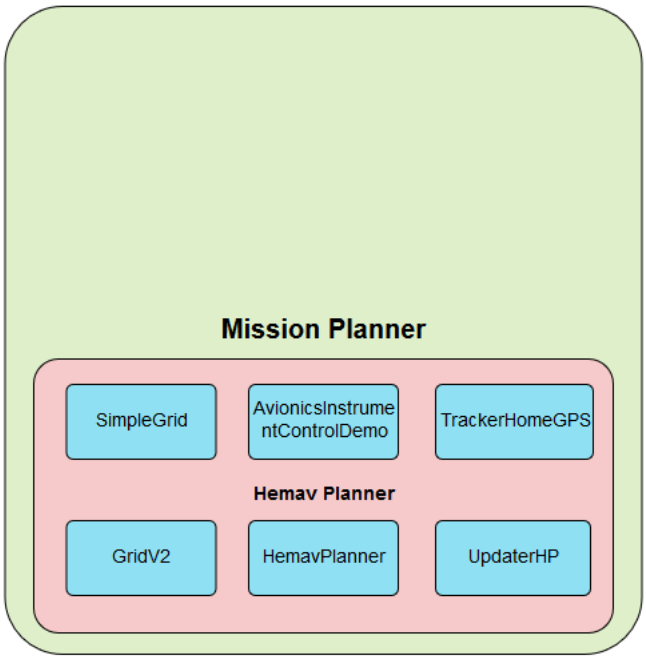

Figura 4.1. Esquema de la arquitectura de plugins del Hemav Planner.

Hemav Planner está construido con varios plugins, pero los más importantes para este proyecto son el GridV2 y el HemavPlanner. En estos dos plugins es donde se desarrolla la mayor parte del proyecto.

Aunque Hemav Planner está diseñado para simplificar la aplicación de Mission Planner, sigue siendo una aplicación de una notable complejidad. Para entender mejor cuales son las labores de mejora que debemos aplicar, es fundamental primero entender el funcionamiento del programa. A continuación, se describe de forma detallada todas las funciones que tenemos a nuestra disposición dentro del entorno de Hemav Planner para familiarizarnos con la aplicación.

La primera ventana que encontramos al iniciar el programa es la pantalla de selección de plataforma.

Figura 4.2. Pantalla principal de selección de plataforma del Hemav Planner.

El funcionamiento del programa depende de la plataforma que elijas. Solo en el caso del ala fija HP2 nos dará la opción de catapulta o bungee.

Una vez seleccionada la plataforma, el programa se divide en tres pantallas principales:

- Datos de vuelo: Seguimiento y control de la misión
- Plan de vuelo: Planificación de la misión
- Configuración: Modificar parámetros y calibraciones de las plataformas.

Datos de vuelo es la pantalla destinada a la operación de vuelo en campo. Incluye la lista de comprobación preflight, el seguimiento de la misión durante el vuelo, la ejecución de acciones de control (iniciar misión, vuelta a casa, aterrizaje emergencia…), la información esencial durante el vuelo y los avisos.

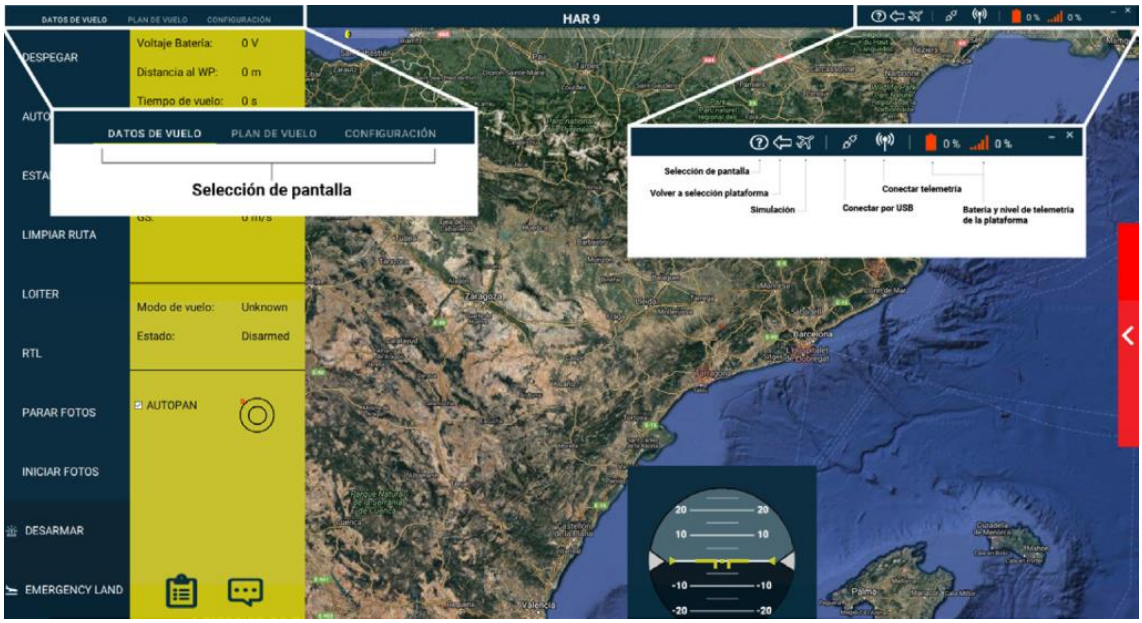

Figura 4.316. Ilustración de la pantalla 'Datos de vuelo' del Hemav Planner.

Botones laterales:

- Auto: modifica el modo de vuelo a automático.

- ARM/Despegar: realiza el armado de los motores en el caso del HAR8 y HAR9, deja la plataforma a la espera de ser lanzada en el caso del HP2 y HP2K.
- Establecer WP: nos permite enviar la plataforma a otro WayPoint distinto del actual.
- Limpiar ruta: limpia las marcas de paso y fotografías hechas de la pantalla.
- Loiter: modifica el modo de vuelo a Loiter. Podemos escoger entre mantener altura o bajar a 50 m. El dron mantiene la altura deseada y se queda estático en el punto cuando se acciona el botón. En el caso de ala fija, da vueltas alrededor del punto en que se ha escogido el tipo de loiter.
- RTL: modifica el modo de vuelo a RTL (return to launch).
- Parar fotos: dejar de hacer capturas fotográficas.
- Iniciar fotos: vuelve a hacer capturas fotográficas.
- Desarmar: modifica el estado a ‘desarmado’ y para los motores. No se permite durante el vuelo.
- Emergency land: activa el modo vuelo “Emergency land” y activa el paracaídas para realizar el aterrizaje de emergencia.

Información de vuelo y avisos:

Tenemos dos zonas de información, lateral izquierdo (fondo verde) donde se muestra información esencial en tiempo real y el bloque desplegable de la derecha donde se indican posibles avisos y alertas para tener en cuenta.

- Voltaje de batería: excepto el HAR8, todas las plataformas dan información del voltaje de la batería. [Voltios]
- Distancia al WP: distancia restante al siguiente waypoint. [metros]
- Tiempo de vuelo: tiempo total del vuelo desde el despegue. [segundos]
- Altura: altura en la que se encuentra la plataforma en relación con el punto de despegue. [metros]
- Próximo WP: nos indica el identificador del siguiente waypoint. [número entero]
- GS: velocidad relativa al suelo (ground speed). [metros/segundo]
- AS: velocidad relativa al aire (air speed). [metros/segundo]
- Modo de vuelo: nos indica el modo de vuelo (Auto, Loiter, RTL, …)
- Estado: nos indica si la plataforma está en estado armado (‘armed’) o desarmado (‘disarmed’).
- Autopan: nos permite decidir si el mapa se auto centra cuando la plataforma está en movimiento.
- Dirección del viento: velocidad y dirección del viento [metros/segundo]

En esta misma columna, tenemos el botón de los requisitos de pre-vuelo y mensajes.

- Pre-vuelo: nos indica las verificaciones que tendremos que hacer en la plataforma. Hay que revisar y presionar en la casilla de verificación de cada apartado.

- MODE CAL (HP2 y HP2K): Modifica el modo de vuelo a manual para poder hacer las calibraciones sin que los servos de las alas trabajen.
- MODE TEST (HP2 y HP2K): Modifica el estado de vuelo a STABILIZE para poder comprobar el buen funcionamiento de los servos de las alas.
- PREFLIGHT CALIBRATION: sirve para calibrar el PITOT del HP2 y HP2k.

- Mensajes: en esta ventana emergente, se muestran todos los mensajes, tanto del programa como los que se reciben de la plataforma. Se aconseja mantenerla abierta durante el vuelo.

PLAN DE VUELO:

En la pantalla de “plan de vuelo” podemos planificar la misión. Se puede planificar manualmente o con la ayuda del programa en base a un polígono dibujado encima de una parcela.

Figura 4.417. Ilustración de la pantalla ‘Plan de vuelo’ con la descripción de las diferentes acciones.

Menu lateral:

En este menú encontraremos las principales acciones para planificar una misión.

- Guardar polígono: una vez dibujado el polígono, podemos guardar un archivo .poly para poder cagarlo en futuras misiones.
- Cargar polígono: podemos cargar un archivo .poly con un polígono. Solo se puede tener un polígono activo, si cargamos un polígono, se borrará el que tenemos en pantalla.
- Activar/desactivar dibujar polígono: con esta herramienta conmutamos entre dibujar polígono y añadir WayPoints.
- Crear misión: nos crea una misión automáticamente dentro del polígono dibujado. Nos abre el survey grid del Mission Planner. No hace falta modificar nada, al escoger una plataforma al inicio, nos carga los datos básicos para un vuelo automático. En el caso de HP2 y HP2K hay que añadir la senda de ascenso y descenso en función de la dirección del viento.

- Buscar localización: Podemos buscar una ciudad, pueblo o zona y nos centra el mapa en ella.
- Cargar KMZ: Podemos cargar un archivo .kmz que nos marcará sobre el mapa la parcela a volar.
- Gráfico de elevación: Una vez creada la misión, podemos ver una gráfica de comparación con los datos de elevación del terreno con los datos de elevación de la plataforma. Es importante recalcar que estos datos son extraídos de Google maps, por lo que son aproximado y solo pueden ser tomar como referencia.
- Limpiar: podemos borrar el polígono, la misión o ambos.
- Guardar WPS: podemos generar un archivo .waypoints o .txt con los datos de los waypoints.
- Abrir WPS: podemos cargar archivo .waypoints o .txt con los datos de los waypoints.
- Leer WPS: una vez conectada la plataforma al programa, podemos leer los WPS que el dron tiene en su memoria.
- Escribir WPS: una vez conectada la plataforma al programa, podemos escribir los WPS en la memoria del dron.

Información geográfica:

En la barra de información geográfica encontraremos:

- Localización de casa: si la plataforma está conectada, podemos geolocalizar el punto de inicio del vuelo.
- Coordenadas de geolocalización de la plataforma
- Si hay un polígono dibujado, nos indica el tamaño en hectáreas.
- Si hay una misión creada con la herramienta automática, nos indica el tiempo de vuelo estimado.
- Información meteorológica de la zona. Nos permite ver la previsión meteorológica del día actual y los cuatro siguientes.
- Fence ALT 230m: permite establecer la altura máxima de la plataforma a 230m en vez de los 150m por defecto. Este parámetro se guarda en la plataforma al escribir los WP. Este cambio quedará registrado en la plataforma cuando escribamos la misión.
- Geolocalización del puntero (ratón).

CONFIGURACIÓN:

En la pantalla de configuración podemos calibrar y cargar parámetros a nuestra plataforma. En este apartado del Hemav Planner se explica detalladamente como hay que realizar calibraciones de los sensores a bordo, así como la configuración de diferentes parámetros, y también nos da la opción de volver a la aplicación original Mission Planner.

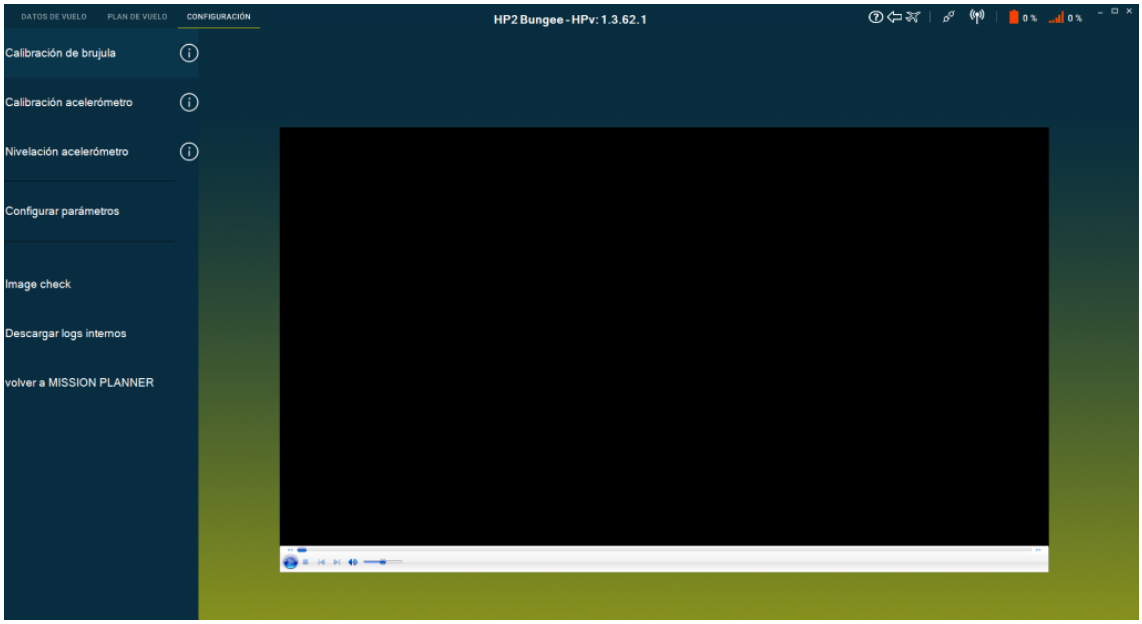

Figura 4.5. Ilustración de la pantalla de 'Configuración' del Hemav Planner.

Esta sección quedará intacta debido a que los procesos de calibración y revisión del dron son tareas difíciles de automatizar y se deberán realizar siguiendo las instrucciones del programa.

Simulación:

Finalmente, en el menú superior derecho encontraremos el botón de simulación (icono de avión). Es muy útil esta herramienta para practicar planificaciones y vuelos ficticios. El procedimiento es muy simple, debemos clicar encima del botón de la plataforma que queremos simular. Previamente, puede ser recomendable subir el número de Sim Speed para acelerar las simulaciones. El resto de los parámetros no son recomendables de modificar. Una vez cargados los parámetros podemos cerrar la ventana de simulación. Una vez estamos conectados al modelo simulado, desde la pestaña datos de vuelo y plan de vuelo interactuaremos con el dron simulado como si de una plataforma verdadera se tratase.

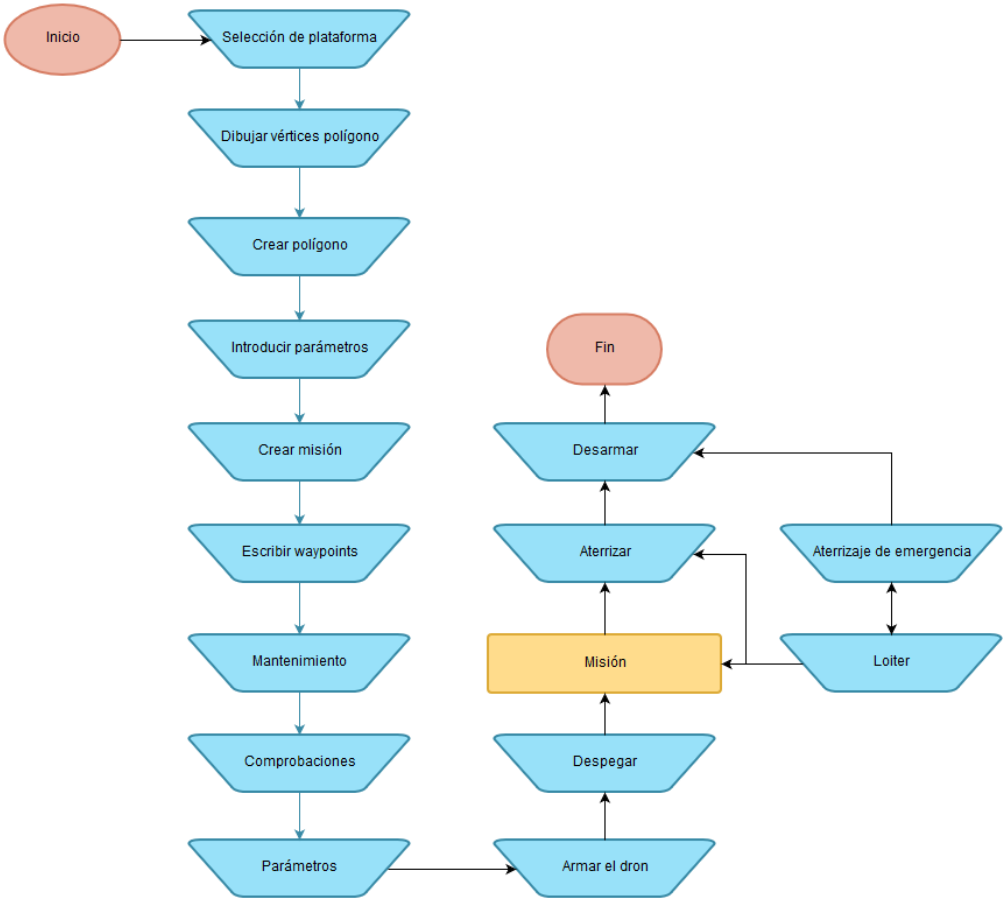

Figura 4.618. Diagrama de flujo de acciones para planificar una misión con el sistema tradicional del Hemav Planner.

4.2. Modificaciones

4.2.1. Reestructuración del código

Como la plataforma de operación será siempre la misma, se fijará el HP2 Bunguee como plataforma seleccionada. Por ello la primera pantalla de selección de plataforma se elimina del inicio de la aplicación y se cargan los parámetros correspondientes al HP2 Bunguee.

Datos de vuelo:

- Auto, Armar y Despegar. Estos dos botones se integrarán en uno para automatizar la tarea.
- Establecer WP. El vuelo será autónomo. Enviar el dron a otro waypoint distinto de la misión supondrá infringir el protocolo establecido para la búsqueda y rescate. Quitaremos la opción para no complicar la misión más de lo necesario.
- Limpiar ruta. Opción prescindible ya que no aporta nada relevante para la misión y aumenta la complejidad de la aplicación.

- Loiter. Opción prescindible ya que el vuelo será autónomo y realizará los loiter según le indique el protocolo.
- RTL. Opción prescindible ya que el vuelo será autónomo y realizará los RTL según le indique el protocolo.
- Parar fotos, Iniciar fotos. Opción prescindible ya que el vuelo será autónomo y realizará las fotos según le indique el protocolo.
- Desarmar. Está opción será automática una vez el dron haya aterrizado.
- Emergency land. Esta función se mantendrá ya que requiere una acción humana para decidir si desplegar el paracaídas.

Plan de vuelo:

- Guardar polígono. Es muy improbable que se realicen dos misiones idénticas en misiones de búsqueda en alta mar, así que esta función es totalmente prescindible.
- Cargar polígono. Análogo al caso anterior, es poco probable que la misión que se pretenda realizar en alta mar pueda coincidir con una ya guardada previamente.
- Buscar localización. En el mar no hay localizaciones concretas ni ciudades, así que esta función es innecesaria.
- Gráfico de elevación. El grafico de elevación es el mismo para toda la superficie del mar, por lo que tampoco tiene sentido incorporar esta función en nuestra aplicación.
- Guardar WPS, Abrir WPS, Leer WPS, Escribir WPS. A priori estas funciones podrían ser útiles de cara a analizar los datos de la misión. Pero no serán necesarias para la aplicación en sí. En cualquier caso, siempre se podrá volver a la versión original de la aplicación cuando se quiera realizar acciones más detalladas de la misión.
- Información adicional de vuelo. Sólo se mantendrá aquella información que resulte imprescindible para la operación de los drones durante las misiones.

Se ha cambiado también la altura a la que se realiza la misión. Este parámetro está fijado por la altura a la que el software de detección de embarcaciones es capaz de funcionar. La altura de vuelo se fijará en 80 metros.

Otro de los aspectos modificados es la separación ente las líneas de la misión. Este parámetro también depende de la altura de vuelo de la misión, ya que lo que se busca es un solapamiento de fotografías de un 10%. De esta forma nos aseguramos de que no fotografiamos la misma zona de forma redundante, pero nos aseguramos un margen por si alguna embarcación se ha movido dentro del área ya fotografiada.

4.2.2. **Nuevas funcionalidades**

Una de las novedades para esta aplicación es la posibilidad de incluir las coordenadas exactas para la misión. Se ha añadido un formulario que emerge después de indicar sobre el mapa de la aplicación un punto donde se quiere

realizar la misión. El formulario permite introducir las coordenadas exactas donde se quiera realizar la búsqueda de navíos.

Esta funcionalidad es necesaria para esta aplicación ya que los operarios de Open Arms no tienen referencias geográficas en alta mar salvo las coordenadas de latitud y longitud. Cuando les llega un aviso de socorro, éste va acompañado de unas coordenadas latitud / longitud que indican la ubicación estimada de la embarcación a la deriva.

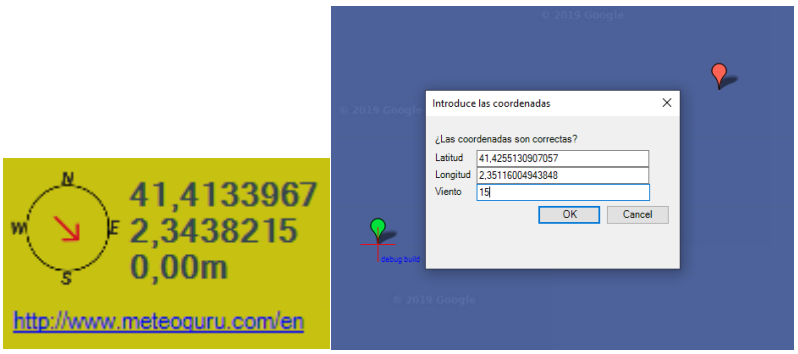

Figura 4.7. Formulario de introducción de las coordenadas de la misión y la dirección del viento con el indicador de dirección de viento.

Por ello, se ha optado por construir la misión dentro de un polígono que se crea en torno a las coordenadas que los operarios introducen en la aplicación.

En el formulario también se puede indicar la dirección del viento. Este parámetro sirve para que el UAV aproveche las corrientes de aire a su favor ya que si se vuela con el viento de cara malgasta parte de la energía en superponerse al viento en vez de en avanzar. Por ello se ha decidido volar en perpendicular al viento con la intención de aumentar el rango.

En el margen inferior izquierdo de la pantalla ‘Plan de vuelo’ se ha situado un indicador de la dirección del viento en unes coordenadas concretes obtenidas de la web de meteoguru.

Otra característica de esta aplicación es que el polígono puede modificar su forma manteniendo el área fija, que era uno de los requisitos de diseño. Esto se pensó así porqué la misión se debe crear aprovechando al máximo el alcance del dron, es decir, el área debe ser tal que permita al dron cubrir la mayor superficie. Esta área viene fijada por la distancia entre el dron y el barco de Open Arms (su estación base) y la autonomía del dron. Por lo tanto, el sistema permite al usuario cierta capacidad de decisión sobre la forma del polígono, pero no sobre sus dimensiones.

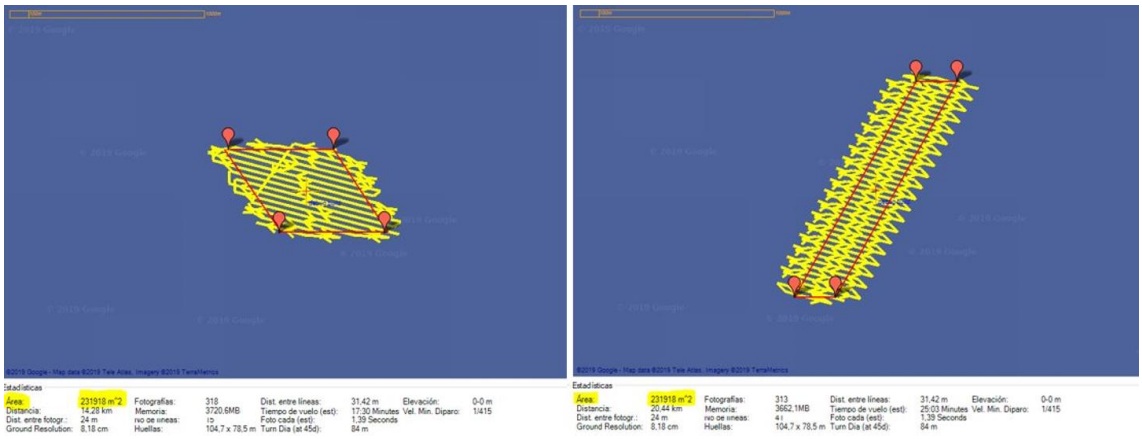

Figura 4.819. Ilustración de cambios de forma del polígono de la misión manteniendo el área sin alterar.

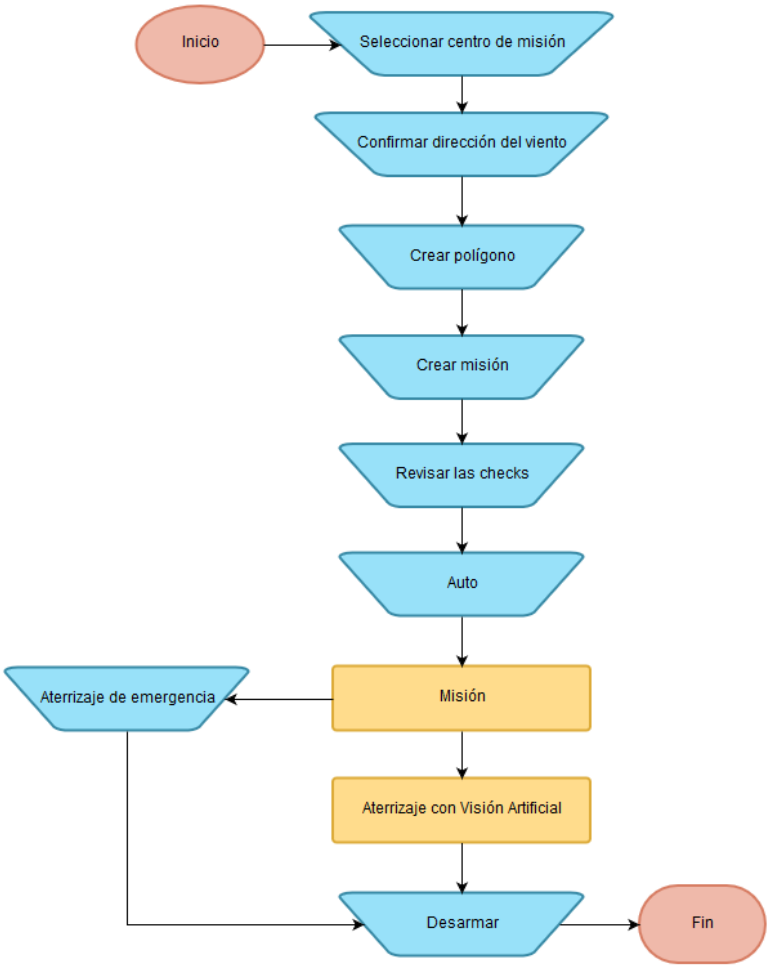

Figura 4.9. Diagrama de flujo de acciones para planificar una misión después de las modificaciones del Hemav Planner. Tras las modificaciones realizadas el número de acciones que el usuario debe realizar antes para lanzar una misión se ha reducido de 15 a 8.

4.2.3. **Test del software**

Para probar el software desarrollado se ha realizado una misión similar a la que se pueda dar en alta mar para el rescate de navíos.

Los parámetros de entrada de la misión son:

Localización: 41,43N – 2.36W.
Viento: 125º con respecto al norte.

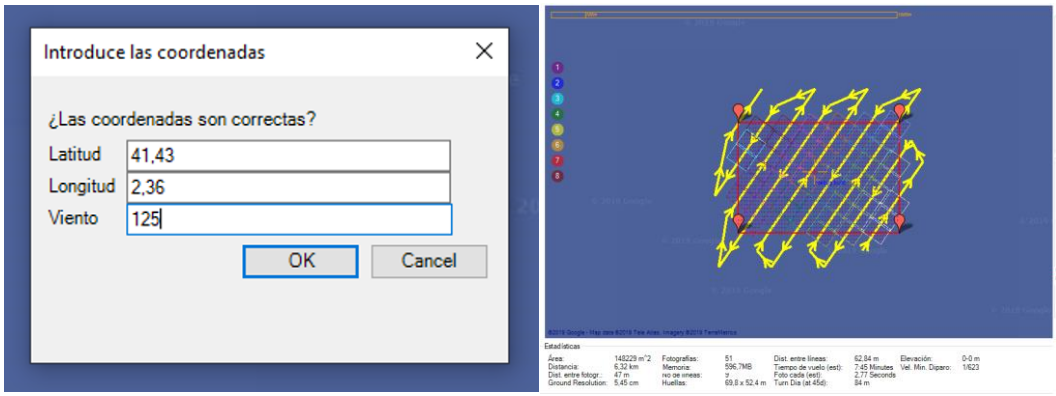

Figura 4.1020. Planificación de la misión e ilustración de la ruta de la misión para la prueba del software.

Después de introducir los datos en el formulario de parámetros de entrada y darle a OK se nos muestra el Grid con las fotografías que se van a realizar durante la misión y la trayectoria que va a seguir el ala fija.

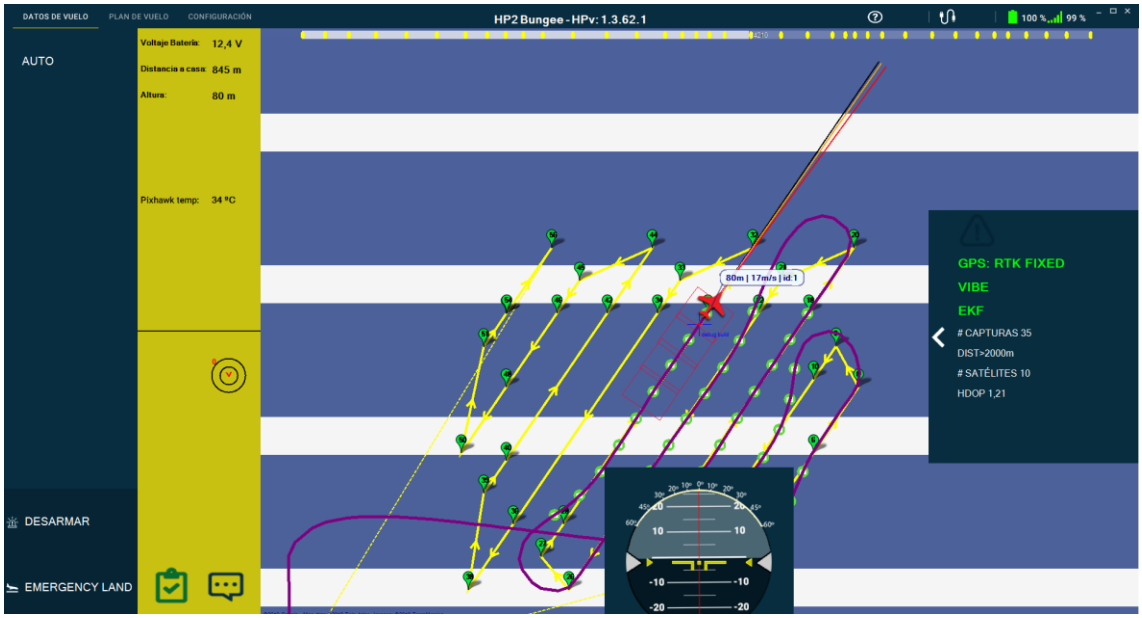

Figura 4.11. Ilustración de la pantalla de datos de vuelo durante las pruebas de la misión.

Tras revisar las indicaciones del formulario del Preflight podemos darle a AUTO y el UAV procede a realizar la misión.

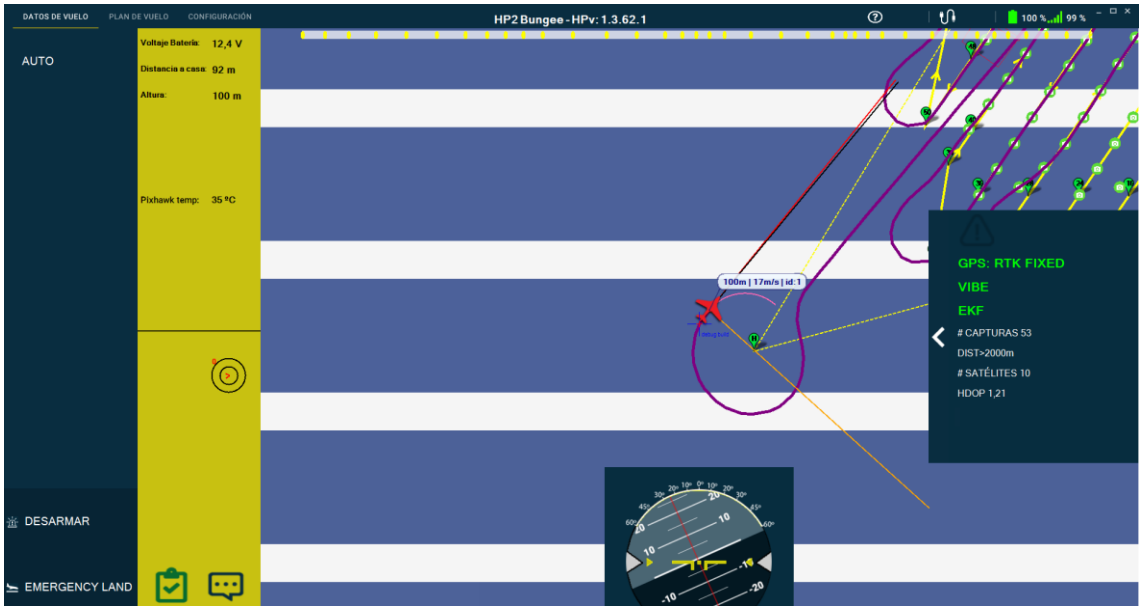

Figura 4.1221. Ilustración de la maniobra final del dron en la fase de aproximación para el aterrizaje.

Para finalizar la misión el dron de ala fija realiza una maniobra circular alrededor del waypoint 'casa', que corresponde a la localización del barco, para comenzar el procedimiento de aterrizaje en el que deberá detectar la red de aterrizaje.

Tras la prueba del funcionamiento de la aplicación podemos comprobar que:

- ✓ La misión se ha creado en torno a las coordenadas indicadas.
- ✓ El tamaño del área se ha creado en función de la distancia y la autonomía.
- ✓ La trayectoria principal durante la misión se realiza perpendicularmente a la dirección del viento introducida.
- ✓ El solapamiento vertical y horizontal de las fotografías es de un 10%
- ✓ El botón AUTO lanza las acciones necesarias para lanzar la misión con sólo ese clic.
- ✓ Tras finalizar la misión el UAV regresa al lugar indicado como casa y realiza una maniobra circular para proceder con el protocolo de aterrizaje.

CAPÍTULO 5. Contribuciones a los otros puntos

5.1. Estudio de la viabilidad para implementar un DGNSS para el aterrizaje

DNSS (GNSS diferencial) es esencialmente un sistema para proporcionar correcciones posicionales a las señales de GNSS. DGNSS usa una posición fija y conocida para ajustar las señales de GNSS en tiempo real y eliminar los errores de pseudodistancias.

Las señales de GNSS que llegan desde los satélites hasta la superficie de la Tierra tienen que viajar a través de las capas de la atmósfera terrestre, por lo que están sujetas a errores. Esto afecta el tiempo que tarda la señal en viajar desde cualquier satélite a un receptor de GNSS causando un error en la posición medida.

Primero, las señales tienen que viajar a través de la ionosfera, la cual está sometida a la continua radiación solar, lo que hace que las partículas se separen y se carguen positivamente. Esta capa de la atmósfera tiene un severo impacto en las señales electromagnéticas que pasan a través de ella. La ionosfera agregará un retraso relativamente grande, el número real dependerá de la ubicación del receptor, la ubicación del satélite, la hora del día, la actividad solar, etc. Esto puede introducir un error de hasta 5 m en la posición capturada.

La segunda capa por la que viajan las señales GNSS es la troposfera. Esta es la sección de clima de la atmósfera, por lo que incluye condiciones tales como nubes, lluvia y rayos. Esto agrega un retardo mucho menor a la señal de hasta 1.5 ns, que puede introducir un error posicional de hasta 0.5 m.

Estos errores son errores aleatorios que fluctúan y no existe manera de medirlos con precisión. Cada error también es específico para cada satélite individual, ya que se ubican en diferentes áreas alrededor del mundo, por lo que sus señales estarán sujetas a diferentes condiciones atmosféricas.

Se puede usar una estación base para proporcionar mensajes de corrección a los retrasos de señal. La ubicación exacta de la estación base se conoce a priori y se usa para modelar el error en sus lecturas de las señales del GNSS. Este modelo se envía al dron para que lo aplique y mejore su precisión. Para realizar esta técnica se asume que el error producido en la medida de la posición de la estación base es el mismo o muy parecido al error producido en las lecturas del sensor GNSS del UAV. Por eso es una técnica de mejora local y se pierde la precisión de este sistema a medida que el receptor se aleja de la estación base.

Además de estos errores, existen errores causados por efectos relativistas debidos a la gran distancia que separa los satélites de la Tierra. También influyen los errores en la precisión de los relojes del satélite y del receptor.

La precisión de DGNSS es del orden de 1 m para los usuarios en el rango de unas pocas decenas de kilómetros desde la estación de referencia, aumentando el error a una ratio de 1 m por 150 km de separación. El Plan Federal de Radionavegación de los Estados Unidos y la Recomendación IALA sobre el Desempeño y Monitoreo de los Servicios DGNSS en la Banda 283.5 – 325 kHz citan el crecimiento de error estimado en 1993 del Departamento de Transporte

de los Estados Unidos de 0.67 m por 100 km desde el sitio de transmisión, pero las mediciones de precisión al otro lado del Atlántico, en Portugal se sugiere una degradación de solo 0,22 m por cada 100 km [13].

5.1.1. **Modelo matemático simple**

Un receptor GNSS identifica las señales de los satélites comparando las réplicas locales de ruidos pseudoaleatorios que emiten los satélites. Una vez identificados los satélites el receptor recibe los datos de navegación de cada satélite que incluyen información sobre la órbita, sus coordenadas, el modelo ionosférico, datos del reloj, bits de paridad, etc. Esto permite al receptor calcular su posición en coordenadas cartesianas mediante la triangulación de las pseudodistancias. La pseudodistancia es la distancia virtual que separa cada satélite del receptor si se tienen en consideración los errores del retardo de los datos provocados por las fuentes de error mencionadas anteriormente.

Una pseudodistancia se puede definir como

$$P = \rho + I + Tr + c(b_{Rx} - b_{Sat}) + \varepsilon_p$$

Donde, P es la pseudodistacia, ρ es la distancia física, I es el error debido a la ionosfera, Tr es el error debido a la troposfera, b_{Rx} b_{Sat} son los errores del reloj del receptor y del satélite, ε_p es el error residual.

Como hemos comentado anteriormente, el DGNSS se basa en las mediciones realizadas en la estación base y envía las correcciones al receptor del dron. La corrección que se debe aplicar se define como la variación de la pseudodistancia que existe entre la medición real (que incluye error) y el modelo que intenta mitigar los errores de transmisión. Si definimos la variación de pseudodistancia entre la estación base y el satélite transmisor como la diferencia entre la pseudodistancia medida y la distancia física real, es decir

$$\Delta P_o = P_o - \rho_o = -I - Tr - c(b_o - b_{Sat}) + \varepsilon_p$$

Podemos calcular la pseudodistancia corregida como la pseudodistancia medida entre el dron receptor y el satélite más la variación de pseudodistancia calculada para la estación base,

$$P_{corr} = P + \Delta P_o = \rho_{Rx} + c(b_{Rx} - b_o) + \varepsilon_p + (I_{Rx} - I_o) + (Tr_{Rx} - Tr_o$$

donde los subíndices Rx y 0 corresponden al receptor dron y a la estación base respectivamente, y la distancia física se define como

$$\rho_{Rx} = \sqrt{(x_{Sat} - x_{Rx})^2 + (y_{Sat} - y_{Rx})^2 + (z_{Sat} - Z_{Rx})^2}$$

Para entender mejor el aumento de precisión de este sistema se ha realizado un estudio que implementa este algoritmo de corrección de posición. Para ellos se han utilizado datos de Ordnance Survey [12] provenientes de estaciones base de Leek y Buxton (Reino Unido). Ambas estaciones están separadas unos 18 km. Esta distancia es similar a la que podríamos encontrarnos en alta mar entre el dron de búsqueda y el barco que se consideraría estación base. Los datos tienen formato RINEX que corresponden a las mediciones de los sensores GNSS tomadas el 18 de julio de 2015.

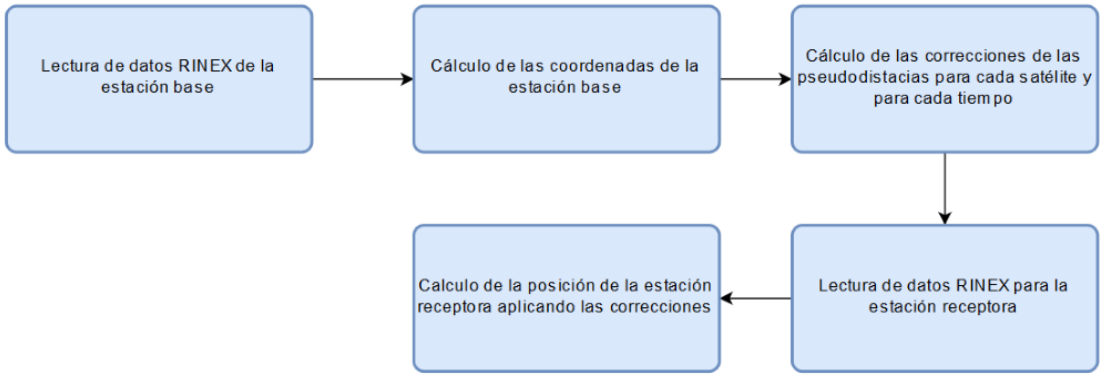

Figura 5.1. Secuencia de procesos ejecutados para el cálculo de las coordenadas con corrección.

Para el cálculo de las coordenadas de los receptores se han utilizado librerías de Mathowrks y archivos subidos por la comunidad de GitLab, junto con código propio utilizado durante la asigantura de NACC.

5.1.2. Resultados y conclusiones del estudio

Las simulaciones muestran que se puede alcanzar un nivel de precisión en torno a 1 metro con el uso de GNSS diferencial.

Tabla 3.1. Resultados de los errores obtenidos con las simulaciones.

	Sin correcciones [m]	Con DGNSS [m]
Error medio	5.63	1.00
Error máximo	14.38	2.30
Error mínimo	0.68	0.06
Desviación estándar	2.36	0.42

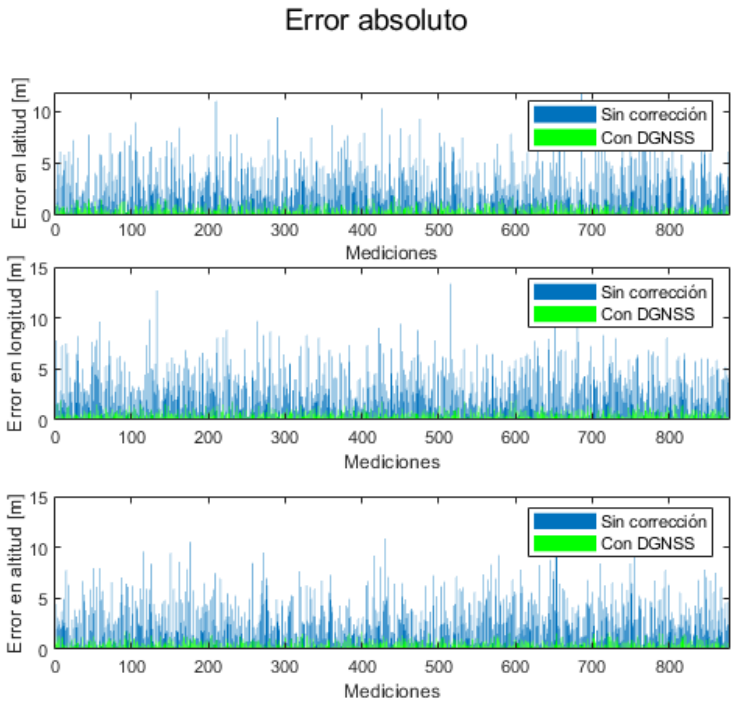

Figura 5.2. Gráfica de errores absolutos en metros para cada medición con correcciones y sin correcciones.

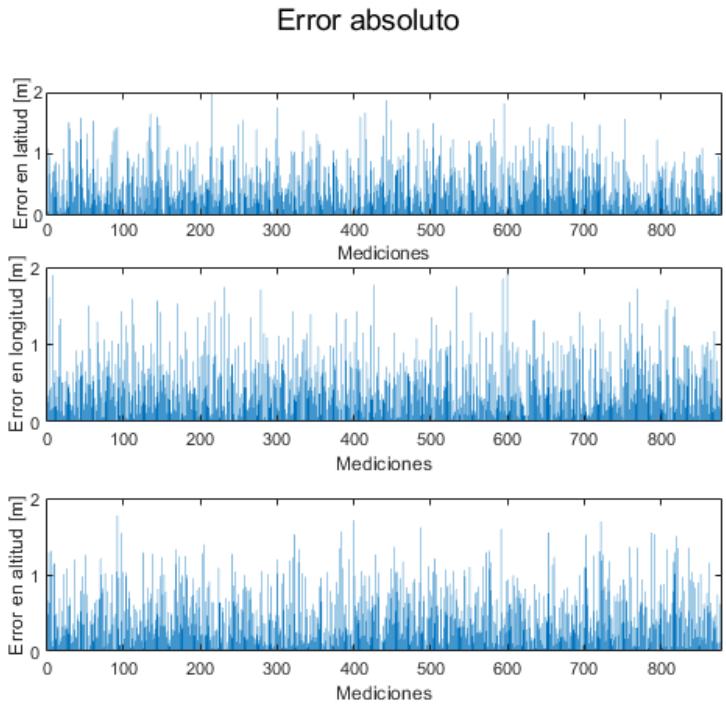

Figura 5.3. Gráfica de errores absolutos en metros para cada medición con correcciones.

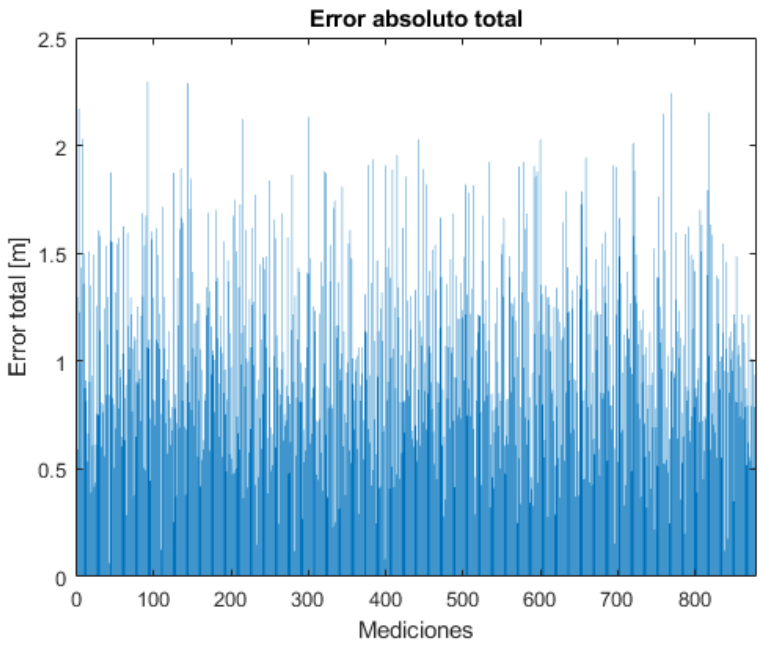

Figura 5.4. Gráfica del error absoluto total (latitud, longitud y altura) en metros para cada medición con correcciones.

Sin embargo, hay que tener en cuenta que en las simulaciones no se ha considerado que el dron opera a una velocidad de unos 12 m/s en su fase de aproximación, y que el sistema de posicionamiento debe actualizar la información lo suficientemente rápido como para que el tiempo de respuesta sea adecuado. Tampoco se ha considerado en ningún momento que, debido al oleaje, el barco está en continuo balanceo y su posición nunca es totalmente fija.

Cabe destacar que para poder obtener las correcciones para el posicionamiento del dron con este software primero hay que procesar los datos del sistema de control de referencia. Cuantas más mediciones de posicionamiento se procesen mejor corrección dará el sistema. Para conseguir una precisión de 1 metro se han procesado 880 mediciones de lecturas de GPS y el tiempo de ejecución del software ha sido de 2 minutos y medio. Esto es un gran hándicap, ya que entra en conflicto con el requisito de velocidad de actualización de los datos.

Teniendo en cuenta que en el peor de los casos podemos llegar a tener un error de precisión de 1.5 metros, que el barco no se encuentra en una posición fija y que está en continuo balanceo, la precisión obtenida podría no ser suficiente a la hora de operar. Además, el sistema todavía no funciona en tiempo real. Por estas razones el DGNSS es un sistema complicado de implementar bajo estas condiciones.

5.2. Estudio del radioenlace por satélite

Este enlace se puede conseguir mediante SPL [9]. SPL es un sistema global de telemetría por satélite para vehículos autónomos controlados por ArduPilot o

pilotos automáticos PX4 que puede utilizar la conexión a Internet TCP/IP o la tecnología de comunicación por satélite ISBD. ISBD es una tecnología de mensajería de gran ancho de banda y alta latencia, sin embargo, es relativamente económica en comparación con otras soluciones de comunicación global. El hardware requerido es muy compacto y ligero.

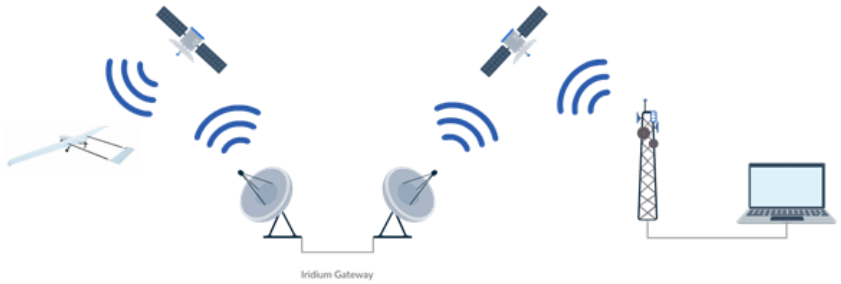

Figura 5.5. Esquema del radioenlace UAV - Estación de tierra utilizando telemetría por satélite de Iridium.

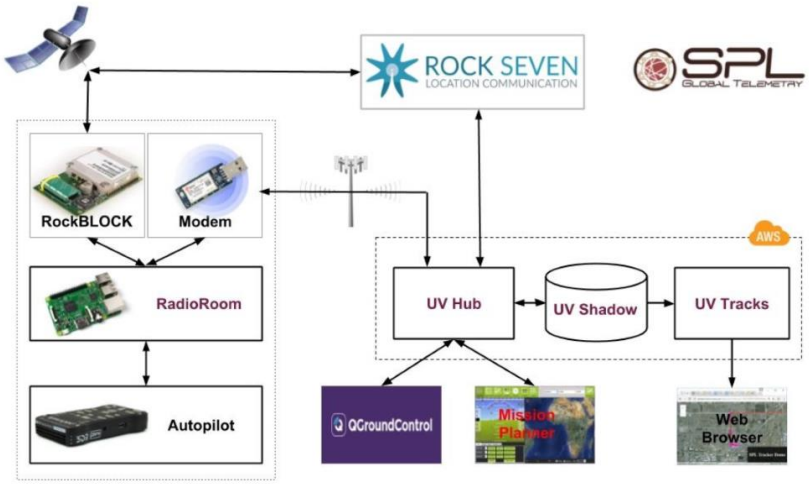

Figura 5.6. Arquitectura del radioenlace por satélite utilizando el software de desarrollo SPL de Rock Seven.

El software de SPL incluye:

- El firmware de UV Radio Room establece la comunicación entre el piloto automático, el módem de Internet y el transceptor ISBD.
- El servidor proxy Hub UV distribuye la comunicación entre las estaciones de control terrestre y los canales de enlace de comunicación del vehículo.
- El servicio web UV Tracks proporciona acceso a los datos almacenados en la sombra del vehículo por UV Hub.

5.2.1. Instalación de UV Radio Room en la Raspberry Pi 3

UV Radio Room es una aplicación para pilotos automáticos basados en MAVLink como ArduPilot o PX4 que ofrece una comunicación entre el piloto automático y un módem de Internet o el transceptor de datos de Iridium (ISBD). Junto con el

servidor UV Hub, proporciona un enlace de comunicación bidireccional entre vehículos no tripulados y estaciones de control terrestre como QGroundControl o Mission Planer.

Para que el UV Radio Room funcione es necesario disponer de una controladora o piloto automático con el firmware de ArduPilot o PX4, una placa base tipo Raspberry Pi, un módulo de comunicación por satélite con un adaptador USB a UART y un modem de internet por satélite. Un ejemplo de módulo de comunicación por satélite es el RockBLOCK Mk2 de Iridium.

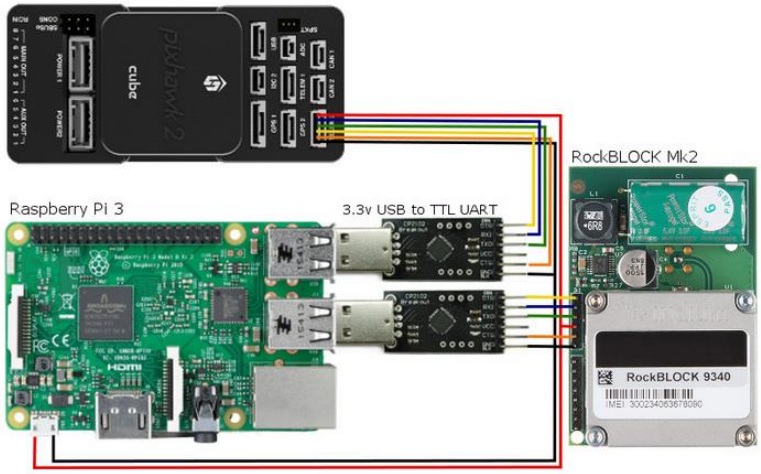

Figura 5.7. Conexión del módulo de telemetría RockBLOCK mk2 con la Raspberry Pi 3 B y la Pixhawk2 Cube.

Una vez se haya hecho la conexión se pueden instalar los paquetes de Radio Room en la Raspberry Pi. Los paquetes se encuentran en GitHub radioroom-2.1.0-raspbian.deb en el enlace

https://github.com/envirover/SPLRadioRoom/releases.

Una vez se hayan instalado se debe configurar las conexiones ISBD y TCP/IP en el fichero /etc/radioroom.conf.

5.2.2. Instalación de los servidores de UV HUB y UV Tracks en la plataforma de servicios en la nube de Amazon Web Service (AWS).

Para realizar esta instalación se debe crear una cuenta en AWS y crear un ‘Amazaon EC2 Key Pairs’. Amazon EC2 utiliza criptografía de clave pública para cifrar y descifrar la información de inicio de sesión. La criptografía de clave pública utiliza una clave pública para cifrar una parte de los datos, y luego el destinatario utiliza la clave privada para descifrar los datos. Las claves públicas y privadas se conocen como ‘Key Pairs’. La criptografía de clave pública permite un acceso seguro utilizando una clave privada en lugar de una contraseña.

También se necesita una cuenta de Rock7 para poder registrarla y vincularla a la clave privada EC2 y registrar la URL del módulo de RockBLOCk que se haya adquirdo como dirección de destino. Además, hay que configurar el archivo

radioroom.conf con la dirección de IP y los puertos que se especifican en la cuenta de Rock7, Rock 7 Core services.

A continuación, se conecta la estación de tierra a la pixhawk y se guardan los parámetros a bordo en un archivo. Luego se conecta la estación de tierra a la IP y al número de puerto especificado y se cargan los parámetros desde el archivo al vehículo para actualizar los parámetros del vehículo.

Este sistema debería implementarse como complemento de un radioenlace directo ya que no garantiza integridad en la transmisión de datos.

CAPÍTULO 6. Pruebas del sistema

6.1. Pruebas en Hemav

A medida que se ha ido desarrollando el sistema se han ido haciendo diferentes pruebas. Las primeras pruebas del software se realizaron en las instalaciones de HEMAV y en el municipio de Abrera.

6.1.1. Detección de red

Una de las pruebas era probar el software de detección de red que se usaría para la maniobra de aterrizaje. La prueba consistía en determinar la distancia máxima a la que el software de detección funcionaba correctamente. Para ello se montó el marco verde de la red y se realizó un vídeo simulando lo que vería el dron a medida que se iba a ir acercando a la red.

Figura 6.1. Captura del vídeo de la red con el procesado de imagen para la detección del marco verde.

Debido a las dimensiones del lugar en las que se realizaron las pruebas, la distancia máxima a la que se pudo grabar sin objetos que obstruyeran la visión fue de 70 metros, los cuales fueron suficientes como para que el marco fuese detectado.

También se realizó una prueba a contraluz para ver si las sombras afectaban la detección. Los resultados mostraron que el software funcionaba incluso con el sol a baja altura.

Figura 6.2. Imagen del marco verde a contraluz.

6.1.2. **Detección de embarcaciones**

Más pruebas se realizaron en Abrera, municipio del Baix Llobregat a unos 30 km del CBL. Ahí se testeó el software de detección de barcos en el mar. Para ello se utilizó una mesa de color blanco de 2 metros de largo y 0.7 metros de largo y se colocó entre la maleza. Seguidamente se sobrevoló esa zona a diferentes alturas.

Figura 6.3. Fotografía aérea realizada con el dron HIR9 durante las pruebas de detección en Abrera.

Las pruebas se realizaron a una altitud máxima de 100 metros, los cuales fueron suficientes para detectar el objeto rectangular blanco. Después de analizar los resultados de las pruebas se pudo escalar los resultados y se estimó que el software de detección funcionaría a 200 metros de altitud.

6.1.3. **Test de protocolos del dron en el mar**

Esta prueba consiste en ejecutar los protocolos diseñados para el dron una vez comience la búsqueda en alta mar. Una de las principales tareas era que el UAV realizara una espiral alrededor del objetivo una vez lo haya localizado.

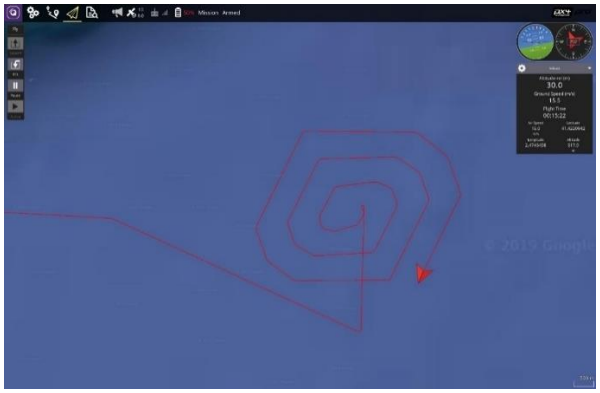

Figura 6.4. Captura de pantalla de la estación de tierra durante la prueba del funcionamiento de los protocolos del dron.

El software de los protocolos se ejecuta en la Raspberry Pi 3 que lee las coordenadas cargadas en la Pixhawk 2 desde el Hemav Planner. Posteriormente simula el escenario donde se detecta una embarcación por el software de detección y reescribe las coordenadas en la Pixhawk 2. La Pixhawk 2 se encarga de dar las instrucciones al dron para que la misión se cumpla según los nuevos parámetros especificados por la Raspberry Pi 3.

También se simuló un barco a la deriva introduciendo coordenadas de una trayectoria con errores en la posición. Esto sometía al sistema a la situación de incertidumbre que se encontraría en el mar para calcular el rumbo y la posición de los navíos. Cuando el dron detectaba una embarcación, comenzaba el protocolo de la espiral que tenía como objetivo encontrar más barcos cercanos. Sin embargo, podría darse la situación en que, buscando otros barcos, el UAV volviese a detectar la misma embarcación. Con tal de evitar esta situación se introdujo un área de incertidumbre que tenía en cuenta la velocidad habitual de los navíos a la deriva, unos 3 nudos.

En las simulaciones comprobamos que la actualización de la misión se realizaba de forma satisfactoria tras la detección de una embarcación.

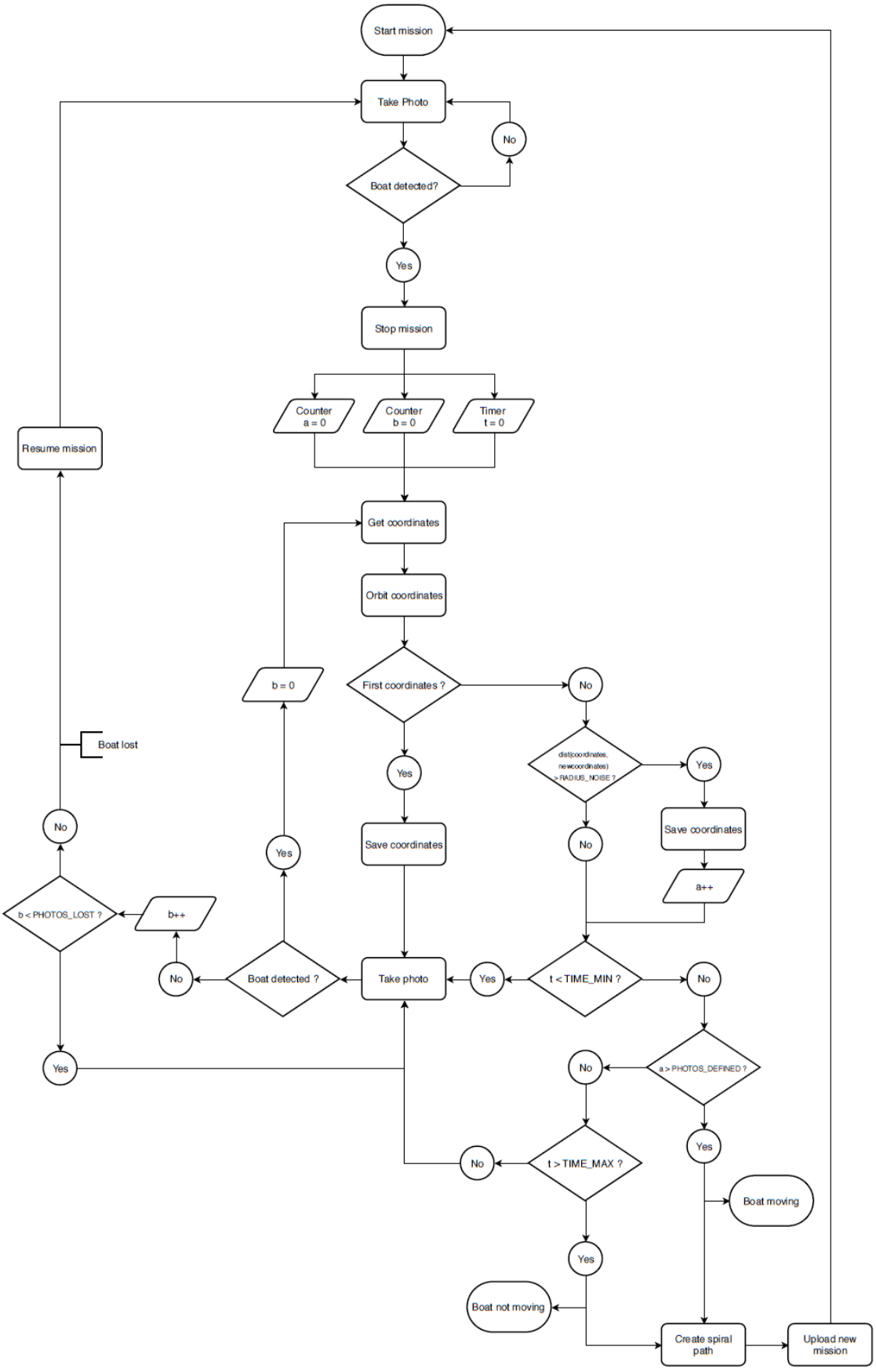

Figura 6.5. Diagrama de flujo de los protocolos del dron para operar de forma autónoma en el mar.

6.2. Pruebas en Burriana

Durante los días 25 y 26 de mayo se realizaron pruebas del sistema en Burriana (Castellón) aprovechando que el barco Open Arms Bilbao atracó ahí antes de partir hacia el puerto de Nápoles. Era muy importante que pudiéramos realizar estas pruebas ya que servían para medir distintas magnitudes del barco y así poder parametrizar nuestro sistema de búsqueda de embarcaciones. También tuvimos la ocasión de probar si los diferentes sistemas que habíamos probado en simulaciones y recreaciones funcionaban en el entorno real. Por último, las pruebas nos permitieron observar el barco para conocer mejor sus características y así poder seguir planificando mejoras para el sistema.

6.2.1. Prueba de telemetría

La prueba de telemetría consistía en medir el alcance del enlace dron-barco que pensábamos instalar. Para las pruebas utilizamos el HIR9, un UAV de 6 rotores que despegaría hasta una altitud de 200 metros en la playa y se mantendría ahí el máximo tiempo posible. Mientras, en el barco, conectamos la telemetría al Hemav Planner donde se indicaba el nivel de señal recibido desde el dron mientras el barco Open Arms Bilbao se iba alejando.

Hubo una complicación y es que la batería del HIR9 permite una autonomía de unos 12 minutos. Después, hay que bajarlo, cambiarle las baterías y volverlo a subir hasta los 200 metros. El problema fue que una vez abajo la señal de la telemetría se perdía completamente debido a la reflexión en el agua y la pérdida de la línea de visión y cuanta más distancia había entre el barco y el dron más costaba establecer el enlace de nuevo. Por este motivo, sólo se pudo realizar la prueba dos veces ya que la tercera vez fue imposible establecer el enlace. La última medición fue un nivel de telemetría del 79% a una distancia de 7,3 NM (13,52 km). Esto se puede considerar un buen nivel de señal ya que se espera operar al doble de esa distancia.

6.2.2. Prueba de altitud para la detección

Para la detección de los navíos a la deriva, si recordamos del capítulo 2.2, era necesario calcular la huella fotográfica de la cámara para procesar las imágenes. Para ello necesitábamos dos parámetros de entrada, la altura a la que vuela el dron y el área que se deseaba detectar. De esta manera calculábamos el número de píxeles que hacían falta para detectar un objeto con el software de detección. La prueba de altitud consiste en realizar el cálculo inverso. Dados los píxeles que sabemos que hacen falta para que el software de detección funcione bien, y el área que queremos detectar, ¿cuál es la altura máxima a la que se podrá detectar las embarcaciones?

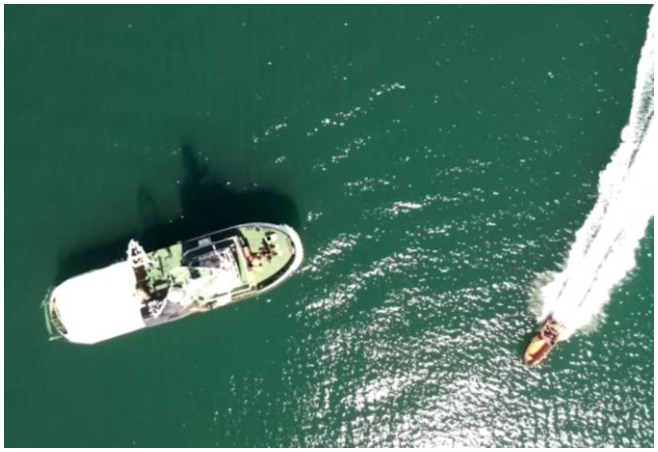

Figura 6.6. Captura del vídeo realizado durante las pruebas de altitud en Burriana.

Para realizar la prueba se sobrevolaron las lanchas auxiliares del Open Arms Bilbao para observar a qué altitud máxima el software de detección era capaz de identificarlos. En las pruebas pudimos ver que a partir de 80 metros de altitud la cámara no tenía la suficiente resolución para detectar las lanchas. Esto queda muy lejos de los 200 metros que se estimaron tras las pruebas realizadas en Abrera. Uno de los factores que han influido en el error de estimación ha sido el reflejo del sol en el agua, que añade mucho ruido a la imagen y es más complicado detectar objetos en ella.

6.2.3. **Prueba de detección de la red**

Para la prueba de detección de red se instaló una red de pesca con un marco verde en la popa del barco. La plataforma utilizada fue el ala fija HP2 que despegó desde la playa. La prueba consistió en realizar varias pasadas justo por encima del Open Arms Bilbao para grabar imágenes que se utilizarán para entrenar el sistema de detección de la red.

Figura 6.7. Captura del vídeo realizado durante las pruebas de detección de red en Burriana.

6.2.4. Prueba de funcionamiento de los protocolos del dron

Esta prueba se realizó para comprobar el funcionamiento del software encargado del vuelo autónomo del dron después de la detección de la embarcación a la deriva. Recordemos que el dron está programado para seguir durante un tiempo los navíos detectados para calcular el área de incertidumbre, la velocidad y el rumbo de la embarcación. Para realizar esta prueba se dispuso del HIR9 y al igual que en las pruebas de altitud de vuelo se sobrevolaron las lanchas del Open Arms Bilbao.

Durante la realización de las pruebas observamos que el dron reaccionaba al detectar los barcos que pasaban por debajo pero no los seguía, se limitaba a girar para perfilarse hacia la dirección del barco y luego perdía el rumbo que debía seguir. Y así una y otra vez hasta que perdía el barco de la visión de la cámara.

Con los resultados obtenidos en esta prueba podemos concluir que el sistema para la búsqueda de las embarcaciones aún no está listo y todavía queda un largo trabajo por delante para poder entregar el diseño final a la oenegé.

6.3. Modificaciones tras las pruebas realizadas

La modificación más importante es el cambio del software de los protocolos del dron en alta mar. Hemos visto que funciona en las simulaciones por ordenador, pero todavía falta mucho trabajo y cambios para implementar este sistema.

La prueba de radioenlace funcionó bien. El kit de desarrollo pMDDL2450 es una opción viable para desplegar este sistema. Sin embargo, la opción de utilizar un radioenlace por satélite se tendrá que implementar, puesto que en el barco Open Arms Bilbao disponen de conexión a la constelación de Iridium y se podría aprovechar la arquitectura SPL, tal y como se muestra en el capítulo 5.

Otra cosa que habrá que mejorar es la cámara utilizada, ya que de cuanta más resolución se disponga, más alto se podrá volar, lo que supone una mayor área inspeccionada, y por lo consiguiente más alcance de búsqueda. Una mejor cámara también aumentaría la distancia a la que se detectase la red de aterrizaje y permitiría realizar la maniobra con más tiempo de antelación.

También nos dimos cuenta de que la estructura que soporta la red de aterrizaje debe ser construida con mayor robustez, ya que el marco que se instaló en la popa del Open Arms Bilbao presentó fallos estructurales que podrían comprometer la fase del aterrizaje.

CAPÍTULO 7. Conclusiones

Este proyecto es un capítulo más del proyecto Freeda que continua en desarrollo y todavía falta mucho para acabar. Aunque no se hayan podido cumplir todos los objetivos iniciales del proyecto en general se han realizado muchos avances hacía el objetivo final.

En este documento se ha intentado explicar el proyecto de forma descriptiva y no se han podido incluir todo el trabajo realizado por el equipo de Freeda y el de Hemav.

De las tareas programadas inicialmente se ha diseñado un sistema capaz de detectar embarcaciones en imágenes, se ha simplificado la estación de control de tierra, y se han hecho bastantes avances en el sistema de aterrizaje mediante visión artificial y el diseño de los protocolos del UAV en alta mar. Aparte de esto, se han realizado multitud de estudios para encontrar distintas formas de enfocar y solventar las tareas de este proyecto. Muchos de estos estudios han sido descartados por complejidad, por no cumplir los requisitos del proyecto o por su alto coste de implementación.

El coste de implementación ha sido en muchos casos el factor decisivo ya que este proyecto se financia con donaciones. Con un mayor presupuesto se puede mejorar el hardware, como las cámaras o equipos de navegación inerciales y el GNSS, se puede mejorar el procesador, incluyendo por ejemplo procesadores con GPU integrada, en definitiva, un mejor hardware se traduciría en mejor rendimiento del sistema.

7.1. Horizonte del proyecto

Este proyecto todavía no está cerca de finalizar ya que todavía hay sistemas que no funcionan correctamente y otros que se pueden mejorar. Además, siempre se puede mejorar el sistema anterior y la idea es seguir desarrollando este proyecto, aunque se haya entregado una primera solución a Proactiva Open Arms.

Tal y como se ha comentado en el capítulo anterior, primero hay que implementar las modificaciones que ha sufrido este proyecto tras las pruebas realizadas en Burriana. Estas modificaciones incluyen mejorar las cámaras para la detección de la red y las embarcaciones con el fin de aumentar la resolución y por consiguiente la distancia de detección. La comunicación por satélite también será clavé para seguir adelante con este proyecto ya que mejorará el alcance del radioenlace.

Además, ya se están desarrollando tareas nuevas de cara a este proyecto. Una de ellas es la integración de un estabilizador gimbal para la cámara de manera que se pueda controlar el ángulo de inclinación de la cámara y evitar los reflejos del sol en el agua.

Otra tarea es diseñar un controlador PID que responda a la detección de la red para el aterrizaje, de forma que la información de la posición de los pixeles correspondientes al marco de la red en la imagen sea traducida en rumbo del UAV y que el PID sea capaz de dirigir el dron hacia la red.

También, como la plataforma seleccionada para las misiones es un ala fija, hay que diseñar un sistema de despegue desde el barco. Actualmente el HP2 se despega con un tirachinas de gran tamaño que necesita unos 17 metros en llano y sin ningún obstáculo para poder despegar. Ahora mismo no existe ninguna superficie en el barco que sea capaz de cumplir con los requisitos de esta forma de despegue, por lo que se deberá pensar en diseñar un sistema que permita al UAV despegar desde el Open Arms Bilbao.

En cuanto a la estación de tierra, el Hemav Planner, hay que realizar un estudio e implementar una función capaz de generar misiones optimizando la distancia recorrida. Actualmente tenemos implementada la función que mantiene un área fija pero la distancia recorrida dentro de ese polígono puede variar con la forma del polígono.

Otra característica que se puede implementar de cara a la futura estación de tierra es una huella con la forma del área inspeccionada en el mapa del Hemav Planner. Esta opción podría ser útil a la hora de realizar misiones múltiples ya que haría la interfaz gráfica del usuario más intuitiva.

Finalmente se estudia implementar misiones de búsqueda a lo largo de las costas ya que muchos navíos a la deriva acaban ahí.

CAPÍTULO 8. Bibliografía

[1]. H. Jin Kim, Mingu Kim, Hyon Lim, Chulwoo Park, Seungho Yoon,Daewon Lee, Hyunjin Choi, Gyeongtaek Oh, Jongho Park, and Youdan Kim, "Fully Autonomous Vision-Based Net-Recovery Landing System for a Fixed-Wing UAV", *IEEE/ASME transaction on mechatronics*, vol. 18, no. 4, pp. 1320 – 1333, august 2013.

[2]. R. K. Sharma, H. B. Hablani, "High-Accuracy GPS-Based Aircraft Navigation for Landing using Pseudolites and Double-Difference Carrier Phase Measurements", *Third International Conference on Advances in Control and Optimization of Dynamical Systems*, Kanpur, India, March 2014.

[3]. Q. Ali, and S. Montenegro, "A Matlab Implementation of Differential GPS for Low-cost GPS Receivers", *The International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 18, no. 3, pp. 343 – 350, September 2014.

[4]. H. Lee, and S. Jung, "Vision - based UAV landing on the moving vehicle", Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, April 2018, DOI: 10.1109/ICUAS.2016.7502574.

[5]. S. Battiato, L. Cantelli, G. M. Farinella, and L. Guarnera, "A System for Autonomous Landing of a UAV on a Moving Vehicle", Dipartamento di Matematica e Informatica, University of Catania, September 2017, DOI: 10.1007/978-3-319-68560-1_12

[6]. K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, "*A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach*", Boston, Birkhäuser, 2007.

[7]. MicroHard, "pMDDL2450 MicroHard". [Online]. Disponible: http://www.microhardcorp.com/pMDDL2450.php , consultada el 16 de mayo de 2019.

[8]. Iridium, "The Iridium System". [Online]. Disponible: http://www.iridium.it/en/iridium.htm , consultada 23 de mayo de 2019.

[9]. Envirover Connected Drones, "SPL Global Telemetry". [Online]. Disponible: http://envirover.com/spl.html , consultada el 3 de abril de 2019.

[10]. Dronecode, "Cube Flight Controller". [Online]. Disponible: https://docs.px4.io/en/flight_controller/pixhawk-2.html , consultada el 23 de febrero de 2019.

[11]. MAVLink, "MAVLink Developer Guide". [Online]. Disponible: https://mavlink.io/en/ , consultada el 14 de mayo de 2019.

[12]. Ordnance Survey, "Government & Business; OS-Net; OS Net RINEX data; Search". [Online]. Disponible: https://www.ordnancesurvey.co.uk/gps/os-net-rinex-data/ , consultada el 2 de junio de 2019.

[13]. Racelogic Support Centre, "VBOX Automotive; General Information; Knowledge Base; How Does DGPS (Differential GPS)

Work?". [Online]. Disponible: https://racelogic.support/01VBOX_Automotive/01General_Information/Knowledge_Base/How_Does_DGPS_(Differential_GPS)_Work%3F , consultada el 15 de marzo de 2019.

[14]. Ardupilot, "Documentation; Mission Planner Home". [Online]. Disponible: http://ardupilot.org/planner/ , consultada el 2 de febrero de 2019.

[15]. Ardupilot, "Documentation; Plane Home". [Online]. Disponible: http://ardupilot.org/plane/index.html , consultada el 2 de febrero de 2019.

[16]. Ardupilot, "Documentation; Autopilot Hardware Options". [Online]. Disponible: http://ardupilot.org/plane/docs/common-autopilots.html , consultada el 13 de febrero de 2019.

[17]. Ardupiolot, "Firmware Site; Firmware; ArduPilot pre-build binaries guide". [Online]. Disponible: http://firmware.ardupilot.org/Plane/ , consultada el 18 de abril de 2019.

[18]. Microsoft, "Microsoft Docs; Documentación de .NET; Guía de C#". [Online]. Disponible: https://docs.microsoft.com/es-es/dotnet/csharp/ , consultada el 6 de junio de 2019.